QL VW

PCT/GB98 / 03440

The Patent Office

INVESTOR IN PEOPLE

09/269624

5620

GB98 / 03440

PRIORITY DOCUMENT

SUBMITTED OR TRANSMITTED IN COMPLIANCE WITH RULE 17.1(a) OR (b)

5640

50XD

VW

The Patent Office
Concept House
Cardiff Road
Newport
South Wales
NP9 1RH

REC'D 20 JAN 1999
WIPO PCT

I, the undersigned, being an officer duly authorised in accordance with Section 74(1) and (4) of the Deregulation & Contracting Out Act 1994, to sign and issue certificates on behalf of the Comptroller-General, hereby certify that annexed hereto is a true copy of the documents as originally filed in connection with the patent application identified therein.

In accordance with the Patents (Companies Re-registration) Rules 1982, if a company named in this certificate and any accompanying documents has re-registered under the Companies Act 1980 with the same name as that with which it was registered immediately before re-registration save for the substitution as, or inclusion as, the last part of the name of the words "public limited company" or their equivalents in Welsh, references to the name of the company in this certificate and any accompanying documents shall be treated as references to the name with which it is so re-registered.

In accordance with the rules, the words "public limited company" may be replaced by p.l.c., plc, P.L.C. or PLC.

Re-registration under the Companies Act does not constitute a new legal entity but merely subjects the company to certain additional company law rules.

Signed

Dated 21/12/98.

Best Available Copy

This Page Blank (uspto)

Patents Form 1/77

Patents Act 1977
(Rule 16)

The Patent Office

07SEP98 E388243-1 D02829
F01/7700 25.00 - 9819389.9

Request for grant of a patent

(See the notes on the back of this form. You can also get an explanatory leaflet from the Patent Office to help you fill in this form)

The Patent Office

Cardiff Road
Newport
Gwent NP9 1RH

1. Your reference — J00040963 GB

2. Patent application number *(The Patent Office will fill in this part)* — 04 SEP 1998 — **9819389.9**

3. Full name, address and postcode of the or of each applicant *(underline all surnames)*

Shell Services International Limited
8 York Road
London
SE1 7NA
United Kingdom

THE PATENT OFFICE L -4 SEP 1998 LONDON

Patents ADP number *(if you know it)* — 750689200 1

If the applicant is a corporate body, give the country/state of its incorporation — United Kingdom

4. Title of the invention — DATA PROCESSING SYSTEM

5. Name of your agent *(if you have one)* — R.G.C. Jenkins & Co.

"Address for service" in the United Kingdom to which all correspondence should be sent *(including the postcode)*

26 Caxton Street
London SW1H 0RJ

Patents ADP number *(if you know it)* — 950001

6. If you are declaring priority from one or more earlier patent applications, give the country and the date of filing of the or of each of these earlier applications and *(if you know it)* the or each application number

Country	Priority application number *(if you know it)*	Date of filing *(day / month / year)*

7. If this application is divided or otherwise derived from an earlier UK application, give the number and the filing date of the earlier application

Number of earlier application	Date of filing *(day / month / year)*

8. Is a statement of inventorship and of right to grant of a patent required in support of this request? *(Answer 'Yes' if:*
a) any applicant named in part 3 is not an inventor, or
b) there is an inventor who is not named as an applicant, or
c) any named applicant is a corporate body.
See note (d))

YES

9. Enter the number of sheets fo[illegible] of the following items you are filing with this form. Do not count copies of the same document

Continuation sheets of this form	
Description	53
Claim*(s)*	13
Abstract	1
Drawing*(s)*	20 + 20

10. If you are also filing any of the following, state how many against each item.

Priority documents	
Translations of priority documents	
Statement of inventorship and right to grant of a patent *(Patents Form 7/77)*	
Request for preliminary examination and search *(Patents Form 9/77)*	
Request for substantive examination *(Patents Form 10/77)*	
Any other documents *(please specify)*	

11. I/We request the grant of a patent on the basis of this application.

Signature R.G.C. JENKINS & CO. Date 04/09/98

12. Name and daytime telephone number of person to contact in the United Kingdom — David C. Musker - 0171 931 7141

Warning

After an application for a patent has been filed, the Comptroller of the Patent Office will consider whether publication or communication of the invention should be prohibited or restricted under Section 22 of the Patents Act 1977. You will be informed if it is necessary to prohibit or restrict your invention in this way. Furthermore, if you live in the United Kingdom, Section 23 of the Patents Act 1977 stops you from applying for a patent abroad without first getting written permission from the Patent Office unless an application has been filed at least 6 weeks beforehand in the United Kingdom for a patent for the same invention and either no direction prohibiting publication or communication has been given, or any such direction has been revoked.

Notes

a) *If you need help to fill in this form or you have any questions, please contact the Patent Office on 0645 500505.*

b) *Write your answers in capital letters using black ink or you may type them.*

c) *If there is not enough space for all the relevant details on any part of this form, please continue on a separate sheet of paper and write "see continuation sheet" in the relevant part(s). Any continuation sheet should be attached to this form.*

d) *If you have answered 'Yes' Patents Form 7/77 will need to be filed.*

e) *Once you have filled in the form you must remember to sign and date it.*

f) *For details of the fee and ways to pay please contact the Patent Office.*

DATA PROCESSING SYSTEM

Background of the invention

The invention relates to a data processing system.

Data processing systems of the type with which the present invention is concerned comprise databases. A database is a collection of structured data for a particular application. The structure of the data is predefined. The data stored in a database may relate to various fields. For example, the data may relate to raw materials used in a process such as a chemical process. The elements each corresponding to a data entry in the database are interrelated with one another in accordance with a predefined structure. Another example of an application of databases is in the field of business information management.

Many businesses or other data processing systems generate a vast volume of data of various types. For example, a business may generate daily files containing records itemising every sale through every outlet; records itemising stock orders and inventories; records itemising taxes paid, and so on. As each process undertaken within an organisation is automated, the volume of data available in electronic form increases.

It would be desirable to collect all such data for analysis. To maintain flexibility for subsequent analysis, it is desirable to store the data in "raw" condition, without omitting or cumulating it (and hence losing information).

This is referred to as "warehousing" the data – i.e. storing it in a data "warehouse" – a large store containing one or more databases of such records.

However, the formats used for sales records differ from those used for inventory or tax records, for example. It is therefore difficult to combine the data from such different sources within an organisation (or across organisations). It might be thought possible to use a common format for all records, but practical difficulties in devising an all-encompassing format in advance, and the inherent redundancy of such a format, make this unsuitable in practice.

Further, existing organisations (especially large organisations) are often necessarily diverse in the way they maintain records. A given product may need a different name, or a different formulation, in different territories, and similarly, an organisation may need to be differently structured in different territories.

Finally, existing organisations (especially large organisations) actually change their structures over time – incorporating new components (with new record systems) and divesting components over time, or changing internal organisational structure.

Thus, an existing data warehouse may be based on a collection of tables, one for each type of transaction for which multiple records are to be stored – for example, a table for daily sales of one product type; a table for weekly sales of bulk products of a different type; a table for monthly inventory

records; and so on. Data in such tables are loaded into the data warehouse from external data sources. The tables are loaded by using loading routines which are specifically designed in accordance with the data structure of the respective external data source from which the tables is loaded, and the data structure of the database into which the tables are loaded. In other words, each loading routine is a unique interface between an external data source and the database. When the structure of one of the records changes, the operator is faced with the choice of opening a new table for the new structure and ceasing to use the old one, or of redesigning the structure of the previous table (or tables) stored for previous transactions and then reloading all such transactions (which may number large numbers of million records).

In the latter case, the tables loaded via the loading routines are then merged on the basis of an integrated data model (i.e. a model which allows combination of the data from different stored transactions, using data reflecting the structure of the organisation and/or the transactions). The integrated data model is pre-structured in accordance with the business requirements, and the format of the source data of the external data sources. The integrated data model is inflexible, i.e. it is designed to contain only data corresponding to its predefined structure. When the business changes, the data model must be redesigned and the data re-loaded as mentioned above.

A populated database may then be used to create an extract which contain selected data from the database and to display the selected data in a

desired format, e.g. in a table, a graph, etc. The desired data is extracted from the database by using a data query routine. Such a data query routine also converts the extracted data into a required data format such that it can be displayed using known spread sheet software, for example.

Figure 1 shows an example of a conventional data processing system. The conventional data processing system comprises three main elements, namely operational systems and external databases 1, a database 2, and data queries 3. The operational systems and external databases 1 contain the data which is to be loaded into the database 2. The data originates from external data sources 4, 5 and 6 each of which uses an individual source data model, as illustrated by the interconnected blocks in databases 4, 5 and 6, for storing the data. . They comprise, for example, multiple sales terminals outputting sales records in predetermined formats; or the sales databases of each regional office of a large organisation.

In order to load the data from the data sources 4, 5 and 6 into the database 2, separate loading routines 7, 8 and 9 are employed respectively. The data in the database 2 is represented in accordance with an integrated data model 10. In order to convert the loaded data from its source data model representation into the integrated data model representation, a separate loading routine 7, 8 and 9 for each external data source 4, 5 and 6, respectively, is required. The integrated data model 10 is specifically designed for the inclusion of data from the external data sources 4, 5 and 6,

the source data models of which are known in advance. If data from an additional external database is to be included in the database 2, a new integrated data model 10 has to be designed.

Data queries 3 are created in order to display a selected set of data from the database 2. Data queries 3 are created by loading the selected data via data query routines 11 and 12 into a suitable display software such as Microsoft Excel (RTM), for example, to display the data, as shown at blocks 13 and 14. On extraction of the selected data from the database, the data is converted into the format required by the display software.

As mentioned, when a database is populated, any changes to the business requirements, for example, on which basis the integrated data model is designed requires a new integrated data model to be created. Such a new integrated data model can be created redesigning the existing integrated data model, defining the (new and old) data sources from which data is to be loaded into the database, and adapting the data loading routines accordingly. The new database may then be completed by loading the data – an operation which may bring the database out of use for some time.

More commonly, however, new entities which reflect the change in business requirements are added to the existing integrated data model without changing the existent data. This can lead to a discrepancy between the "logical" data model of the data warehouse and its actual physical realisation.

Such systems encounter disproportionately high maintenance costs as new subject areas (entities) have to be added to the warehouse, or the entire design has to be changed completely to reflect the changed external business environment. Maintenance costs per year of 25% to 100% of the initial development costs are not uncommon. By way of comparison, in transaction processing systems the annual maintenance costs are typically 10% to 15% of the development costs.

This high ongoing cost for a data warehouse is a major contributing factor to why many data warehouse projects do not sustain existing business requirements. Organisations simply may not appreciate what level of investment can be necessary to deal with reflecting business and chronological changes. Indeed, with conventional data warehouse designs, it is questionable as to whether these can ever be satisfactorily reflected.

Accordingly, it is desirable to provide a data processing system which addresses one or more of the above disadvantages.

Summary of the invention

According to one aspect of the present invention, there is provided a data processing system, comprising: processing means for generating a data model in accordance with a data structure, the data model being adaptable to represent a change in the data structure; and storage means for storing the data in accordance with the generated data model.

According to another aspect of the invention, there is provided a data processing system, comprising: processing means for generating a data model representative of data of a first structure, and for adapting the data model to represent also data of a second structure; and storage means for storing data in accordance with the data model.

Accordingly, it is possible to include data of widely variable structure in the data processing system. This can be done by adapting the data model to a change in the structure of the received data. It is no longer necessary to fully predefine the data model because the data model is adaptable to new and unanticipated requirements. Thus, the data processing system is highly flexible and can be adapted to any changes in the external requirements at any desired point in time.

Preferably, the data model includes information representative of the time of change in the structure of the received data, or of the time of adaptation of the data model. Accordingly, not only does the data processing system support the inclusion of data having a different structure, but also the inclusion of information reflecting when the data model was changed, i.e. when the structure of the received data has changed.

Thus, the data processing system is capable of storing historic information. For example, if the data processing is used for business information management purposes and the underlying data sources are changed at an arbitrary point in time (due to a business reorganisation), the

data processing system stores data reflecting that change. Thus, not only the data itself (representing the business activities) before and after the change may be stored, but also the change of the data structure (representing the business organisation) over time. By contrast, traditional systems only represent a snapshot of the business requirements valid at the time when the system was designed. This makes it difficult to store historic information, which may well require as much analysis as to load the data itself. In traditional systems, therefore, historic information is discarded due to the extra analysis required.

In one embodiment, the stored data comprises: transaction data representative of one or more measures which are determined relative to one or more references; reference data representative of said one or more references; and metadata descriptive of the transaction data and the reference data. The metadata may define hierarchical associations between classes of the reference data.

The stored data may comprise a number of elements of reference data, each element of reference data comprising information which defines an association with one or more other elements of reference data. Each element of reference data may further comprise information representative of a first period of validity of a defined association. The information representative of the first period of validity comprises a start date of validity and an end date of validity.

The one or more measures each may be associated with one or more units. The associations between the one or more measures and the one or more units may be associated with a second period of validity. The second period of validity may comprise a start date of validity and an end date of validity.

The stored data may comprise a number of items of transaction data, each item of transaction data being associated with a date of transaction.

The metadata may define associations between classes of reference data and the one or more measures, the associations between the classes of reference data and the one or more measures being representative of classes of transaction data.

The data processing system may also comprise first interface means for receiving data of any structure from a data source for storage in the data processing system. Also, the data processing system may comprise second interface means for outputting data from the storage means in a required format.

Accordingly, it is unnecessary to use different loading or outputting routines for different data structure requirements. Rather, the interface means are generally applicable and reusable in accordance with the used or required data structure.

Other aspects and preferred embodiments of the invention are as described hereafter, or as detailed in the accompanying claims.

It should be noted that, whilst the provision of the ability to change the data over time (for example by the inclusion of stored validity range data) is one inventive feature of the disclosed embodiments, other features of the disclosed embodiments may be used separately of this aspect and protection is sought for such other features of the invention in isolation, as well as in combination with the foregoing aspect of the invention.

Brief description of the drawings

An embodiment of the invention will now be described, by way of example only, with reference to the accompanying drawings in which:

Figure 1 shows a schematic illustration of a conventional data processing system;

Figure 2 shows a schematic illustration of a data processing system in accordance with an embodiment of the invention;

Figure 3 shows a schematic illustration of the types of data used in the data processing system in accordance with the embodiment of the invention;

Figure 4 shows a schematic illustration of a first type of data used in the data processing system;

Figure 5 shows a schematic illustration of the data fields used in the first type of data;

Figure 6 shows a schematic illustration of a second type of data used in the data processing system;

Figure 7 shows a schematic illustration of how the second type of data is structured:

Figures 8a and 8b show a schematic illustration of the data fields used in the second type of data;

Figure 9 shows how the second type of data is stored in the data processing system;

Figure 10 shows the steps taken to initialise the data processing system;

Figure 11 shows an exemplary classification of products relating to a use of the data processing system for business information management;

Figure 12 shows a first data structure used to represent a hierarchical data classification;

Figure 13 shows a second data structure used to represent a hierarchical data classification;

Figures 14 to 16 show an example of a business re-organisation supported by the data processing system;

Figures 17a and 17b show output displays produced by the embodiment at differing levels of hierarchical detail of a product classification;

Figure 18 is a further screen display produced by the embodiment and showing the hierarchies of which a given product is a member;

Figure 19 is an annotated screen display produced by the embodiment to input the parameters for data extraction;

Figure 20 is a diagram showing schematically the subprograms present in the embodiment;

Figure 21 is a flow diagram showing schematically the process of amending reference data stored in the embodiment; and

Figure 22 is a flow diagram showing schematically the process of extracting data in the embodiment.

Detailed description of the drawings

The Data Processing System

Figure 2 illustrates a data processing system 20 in accordance with an embodiment of the invention. The data processing system 20 is implemented on a server in a computer network The sever comprises a large storage device 21 (e.g. a high capacity disk drive or array of disk drives), a processor 211 (e.g. an Intel Pentium™ processor) arranged to read and write data thereto, and to perform the processes described hereafter. Under the control of programs loaded into a random access memory 212. Referring to Figure 20, the programs comprise a transaction data loading programs; a reference data loading program; a data browsing program; a data amending program; a querying and outputting program; and operating system (such as Unix™); a graphical user interface (GUY) such as X-Windows or Windows™; and a communications program for communicating with external devices. Acting as

a container for the data structures described herein is a database program (e.g. Oracle™) providing a database file stored on the storage device.

The server 21 is connected to a plurality of workstations 22a, 22b and 22c through connections 23a, 23b and 23c, respectively (for example forming part of a Local Area Network (LAN)). Also, the server 21 is connected to databases 24a and 24b through connections 25a and 25b, respectively (for example forming part of a Wide Area Network (WAN)). The databases 24a and 24b serve for collecting external data (illustrated by arrows 26a and 26b) for storage in the data processing system 20. The data is loaded into the data processing system constantly or at regular intervals.

For example, the data processing system may be used in the field of business information management, and the databases 24a and 24b may be used for collecting and storing business transaction data (i.e. data representing the business' activities). Depending on the size of the business, the amount of the data collected by databases 24a and 24b may be considerable, e.g. up to millions of transactions per time interval.

The data processing system 20 comprises interface means (in the form of loading programs and an associated user interface for defining parameters thereof) for receiving data from the databases 24a and 24b without the need for the user to write a specific data loading program.

The data, when loaded and stored in the data processing system, is classified in accordance with a generic data model. This data model is described in more detail below.

The stored data can be accessed and loaded by the work stations 22a, 22b and 22c. However, due to the potentially vast amount of data stored in the data processing system, the data is not normally transferred to the work stations 22a, 22b and 22c as a whole. Rather, the user of any of the work stations 22a, 22b and 22c defines a data query in order to load only data which is relevant to her/him. Such a query causes the data processing system to retrieve the requested data and to transmit it to a workstation in a required data format. This process will be described below in greater detail.

Types of Data Used in the Data Processing System

Figure 3 shows a schematic illustration of the three types of data used in the data processing system for storing data. The data is classified as reference data, transaction data and the metadata. All three types are held within particular defined tables within an available database program (for example, Oracle™) in the storage device of the server 21.

The data processing system uses transaction data as indicated at box 30, reference data as indicated at block 31, and metadata as indicated at block 32. The transaction data 30 comprises fields 33 for holding numeric values, and fields 34 holding pointers to elements of the reference data. These three types

of data are described below in more detail in the exemplary environment of business information management. It is to be understood that the three types of data could as well be used for representing data relating to, for example, an industrial process.

The reference data comprises a plurality of records defining respective business entities, and the associations between them. A business entity is an identifiable thing within the business to which costs, sales and other information resulting from individual business transactions (held in the transaction data) can be related. Examples of business entities include names of brand managers, periods of sale, etc.

The transaction data comprises data items (values) relating to (business) transactions. A data item represents an individual value. Examples of data items include "15 litres", "25(USD)", etc. An example of a business transaction is "the sale of 1500 Litres of substance x to customer y on date z". A transaction will have a number of values (data items) associated with it which can be related to a set of Business Entities. In fact, each transaction is typically an operation involving one or more such entities (for example, the sale of a product from a first entity to a second entity).

The metadata defines the classes of business entities ("CBE"s), corresponding to classes of reference data in the business context), transactions and data items. It thus indicates the possible relationships (for example, hierarchies) between business entities.

A class of business entity defines a type of business entity. Examples include "year", "country", “company”, “branch”, “product family” or “product”. A class of transaction defines a type of business transaction. Examples include "sales orders", "purchase orders", "market surveys", etc. A class of data item defines a type of data item (also known as a measure). Examples include "sales volume", "net proceeds", etc. A measure may be defined as a stored formula calculated from one or more other measures.

The classes of entities therefore represent dimensions across which the measures held in the transaction records can be analysed, summarised and plotted. For example, sales volume, price volume or cost can be analysed across the “customer” dimension, or the “country” dimension, and so on, if the necessary data is held in the metadata for such analysis. Some data items dimensions (e.g. volumes) can be summed up across several dimensions, while others (e.g. temperatures) can typically only be analysed over one. Many of the entities correspond to parties to transactions within the transaction data (e.g. the buyer or the seller, or parts thereof). In addition to the classes of business entity, one other dimensions over which data is summarised is time.

Transaction Data

Figure 4 shows a schematic illustration of a particular type of sales transaction. The transaction (indicated at 40) is associated with one or more

measures. These measures are indicated at 41 and include "Volume" and "Manufacturing costs". The measures, in turn, are measured against one or more dimensions. These dimensions correspond to classes of business entities. In Figure 4, these dimensions are "Delivery Date" at box 42, "Delivery Point" at box 43, "Packed Product" at box 44, and "Sales Representative" at box 45.

Figure 5 illustrates a transaction dataset as stored in the data processing system. The transaction dataset comprises various fields for holding the data in accordance with the schema of Figure 4. Fields 50 to 53 hold pointers pointing to the dimensions associated with the transaction. The term "pointer" here is used to represent the function of fields 50 to 53. The pointing is carried out by storing identifiers in fields 50 to 54 indicating the database index code of the reference data elements (dimensions) to be pointed at.

In particular, field 50 holds a pointer pointing to the reference data record for the particular sales representative associated with that transaction, field 51 holds a pointer pointing to the delivery point associated with that transaction, field 52 holds a pointer pointing to the packed product the subject of that transaction, field 53 holds a pointer pointing to the delivery date associated with that transaction, and field 54 holds the transaction date.

The transaction date is used for handling time-variant entries into the data processing system as is described below.

Field 55 holds a numeric value representing the volume of the transaction, and field 56 holds a pointer pointing to the record holding details of the unit in which the volume is measured. Similarly, field 57 holds a numeric value representing the manufacturing costs, while field 58 holds a pointer pointing to the record holding details of the unit in which the manufacturing costs are measured.

Thereby, each measure is associated with a unit in which the measure is represented. Since a stored measure is invariant (i.e. invariant numeric values), the association of that measure with a unit is invariant. In other words, each measure is associated with a single unit for all time.

However, a stored measure can be displayed in a selected unit rather than only in the associated unit where suitable conversion processes 9 e.g. multiplication by a constant to convert between two units of weight) are stored within the system. If the selected unit is different from the associated unit, then the stored measure is converted into the selected unit before display. Where the conversion rates change frequently (for example, currency exchange rates), the conversion rates are stored as daily transaction data records.

It is to be noted that the data processing system supports multiple definitions of how transaction data is measured against dimensions. It supports measurement of disparate sets of transaction data against disparate sets of dimensions, respectively. However, it also supports measurement of

multiple sets of transaction data against shared sets of dimensions, or against a combination of shared and disparate sets of dimensions.

The transaction data, as indicated above, forms multiple different "sections", each section corresponding to a different defined transaction type; for example, a section for product sales, a section for bulk sales, a section for inventory records and so on. Within each, periodically, new transaction records are loaded from the external data sources as discussed above, so that the total numbers of transaction records will become large.

Reference Data

As indicated in connection with Figure 3, the second type of data used in the data processing system, the reference data, describes dimensions against which transactions are measured. In the field of business information management, these dimensions are often referred to as "Business Entities". Examples for reference data, as given above, are date of sale, delivery point, etc.

Any dimension or reference data item may be related to other items of reference data. For example, the delivery point can be a sub-group of a geographical area. The geographical area may be a sub-group of a country, and so on. These interconnections are called associations.

By defining associations between elements of reference data, a hierarchical (or other) structure of reference data can be formed. An example

is given in Figure 6. The saleable product at box 61 is branded as a product name as indicated at box 62, which in turn is a member of a product family (box 63), which product family is managed by a brand manager (box 64). Thus, the reference data record for the saleable product record (a member of the saleable product class of entity) points to an association record which also points to the product family record (a member of the product family class of entity) and so on. Any of the dimensions shown in Figure 4 can be classified in a similar way, if the associated class of entity record indicates this is possible.

It is to be noted that though the above discussion relates to a strictly hierarchical data structure, non-hierarchical relationships (i.e. many to many associations) can also be represented in this way.

Figure 7 illustrates how reference data is modelled in the data processing system. Boxes 71 to 74 represent the same reference data elements as shown in Figure 6. The relationships between the reference data elements 71 to 74, illustrated by arrows in Figure 6, are represented by boxes 75 to 77. The records storing data for these relationships are called "associations" herein.

Both the reference data elements and the associations represent items of data ("objects") stored in the data processing system. This is illustrated by Figures 8a and 8b.

Figure 8a shows a reference data element containing fields 80 and 81. Field 80 holds the actual reference data entry such as the name of a brand manager. Field 81 holds a unique identifier which is used to reference the data element by use of a pointer in a transaction data item as explained above.

Figure 8b shows an association data element comprising four data fields 82 to 85. Fields 82 and 83 contain a period of validity consisting of a start date and an end date, respectively. Fields 84 and 85 hold identifiers which define an association of one reference data element with another reference data element. Each of the identifiers 84 and 85 corresponds to a respective different identifier in a reference data element (see field 81 in Figure 8a). For example, association 75 of Figure 7 contains the identifiers of the brand manager 71 and the brand family 72.

The period of validity is representative of when an association was formed and when an association ceased to exist (if at all). In the example of Figure 6, "Paul Bishop" is shown as the present brand manager of the "Shell Helix" product family. If, due to a business re-organisation, another brand manager is appointed to replace Paul Bishop, a new association is created between the "Shell Helix" product family and the newly appointed brand manager. The association data of the previous association, however, is retained in the data processing system.

In other words, after the business re-organisation, the data processing system stores data reflecting the association of Paul Bishop with "Shell Helix"

from a start date (date of appointment of Paul Bishop as brand manager of "Shell Helix") to an end date (date of appointment of Paul Bishop's successor) and, additionally, data reflecting the association of Paul Bishop's successor from a start date (date of his/her appointment) up to present (no end date). Thus, the data processing system retains historical information representative of the business organisation at any point in time.

In the above discussion, periods of validity are mentioned in connection with associations between reference data elements. However, it is to be noted that an object stored in the data processing system may include information relating to its period of existence.

In the above example, Paul Bishop may have retired and therefore "cease to exist". Accordingly, not only associations of Paul Bishop with other reference data elements, but also the reference data element itself may hold a start date (Paul Bishop's appointment in the business) and an end date (Paul Bishop's retirement).

Figure 9 illustrates a preferred additional feature of this embodiment. In which the reference data (i.e. reference data elements and its associations) is additionally stored in the data processing system in so-called "mapping tables".

Each mapping table comprises rows in the format shown in Figure 9. Fields 90 and 91 hold a start date and an end date, respectively. These dates define a period of validity of one of the associations discussed above.

For example, fields 90 and 91 hold the dates defining the validity of the association of Paul Bishop with “Shell Helix”. Accordingly, the name “Paul Bishop” is stored in field 92 while “Shell Helix” is stored in field 93. In addition, the map table row comprises fields 94 and 95 containing reference data elements which are also included in the hierarchical structure, namely the product name 94 and the saleable product 95 of the illustrated example (see Figure 6).

Accordingly, the data processing system in the illustrated embodiment stores one row for each pair of start and end dates. By doing this, the complex data structures are converted into simple tables which represent the data structure hierarchies (corresponding to the business organisation) at any one point in time. The manner of use of such tables is discussed below.

Metadata

The third type of data, the metadata, can be described as "data about data". Metadata is descriptive of the reference and transaction data, the associations between elements of reference data, and the measures associated with transactions. More specifically, the metadata provides a classification of the reference data, the transaction data and the measures. Such a classification is defined by a user of the data processing system. The user can define different classes of each reference data, transaction data, and of measures.

The purpose of the metadata is to provide a catalogue of what information is contained in the data processing system, to find data in the data processing system, and to guarantee that the transaction data and the reference data is consistent with the business definitions. The metadata is used to query data for display, and for loading data from external databases.

A class of reference data can be understood as a stored record acting as a holding place for reference datasets. For example, the name of a brand manager is an element of the class "Brand Manager". The former is a reference data element whilst the latter is a class of reference data. Similarly, a class of transaction is a holding place for transaction datasets. For example, "Sales" is a class of transaction including the elements "Export Sales" and "Inland Sales". Also, a measure is a holding place for the actual values in which the transaction data is measured which is associated with a specific unit.

The metadata defines the valued units that can be used for any measure. For example, a measure "Cost of Manufacture" is associated with either a single unit such as "Pound Sterling" or "Deutschmark", or with multiple units so that each actual value can have a different unit. These associations define which units are valid for a measure and are used for validation of loaded transaction data, and for setting default units. The associations can be changed over time.

Also, the metadata defines associations between classes of reference data. An association is defined as a record indicating a parent class of reference data and a child class of reference data. For the parent class of reference data, the association is a downward association, while it is an upward association for the child class of reference data.

All associations are defined as having rules of cardinality allowing an associations to be set as either mandatory, optional or principle. In the case of a mandatory association, the child class of reference data cannot exist without having a parent class of reference data. In the case of an optional association, the child class of reference data can exist without having a parent. A principle association applies for a child class of reference data which has multiple upward associations. One and only one may be defined as the principle association.

These associations, defined as metadata, are used when loading reference data so as to be able to verify whether the loaded data corresponds to the defined data model. As mentioned above, the data processing system may thereby use a more generic interface program for loading transaction data of several types of transaction without the need to write specific program code for each. Rather, the loaded data (reference data and transaction data) is verified for consistency with the metadata definition of the transaction and reference data. Inconsistent data records are rejected and temporarily stored in a holding area for correction, re-validation and re-submission.

Initialisation process

The above types of data are stored in the data processing system using a table for holding reference data and metadata, and one or more tables for holding numeric values (representing the measures) and pointers (identifiers) to elements of the reference data.

Initially, the data processing system does not contain any data, and no data model is defined. Accordingly, the data processing system has to be initialised. This is illustrated in Figure 10.

Initially, the metadata has to be defined (i.e. input by the user) in order to provide a data model on which basis reference and transaction data may be loaded into the data processing system.

At step 100, classes of reference data are defined. A class of reference data represents a holding place for reference data entries (of that class) in the data processing system. A new class of reference data is defined by a user by entering a desired name for that class of reference data.

Subsequently, the user may define an association of that new class of reference data with another class of reference data. To do this, the user defines another new class of reference data and the defines the association between the two new classes of reference data. The user has to define the kind of association, i.e. whether the other class of reference data is a "parent"

or a "child" of the previous class of business entity, and whether it is hierarchical or non-hierarchical.

For example, the first new class of reference data may be "Country". Then, another class of reference data "District" is defined. Since a country covers several districts, the class of reference data "Country" is defined as the parent of "District". The user may define further child or parent associations with "Country", "District", or any other defined class of reference data. District could also have a second association with other classes of reference data used to classify district, e.g. climatic conditions, altitude ranges, type of area (rural, suburban, city). These could be defined as hierarchical or non-hierarchical.

In this embodiment, a plurality of common, predefined classes of entity are provided for selection by the user, together with typical relationships therebetween; for example, geographical entities, companies and branches thereof and so on. The user is free to add newly defined entities additionally or alternatively to these.

For this purpose, a graphical user interface (GUI) program is provided which causes the display on the workstations 22 of a screen showing the existing entity classes and their associations, and allowing the input of data, via a mouse and/or keyboard of the workstations, defining new entities and associations.

Also, the user has to define one or more naming schemes (also referred to as descriptors) which are associated with a class of reference data. A naming scheme normally is a code identifying an element of reference date. For example, a country code is used to represent a country. In this case, "Country Code" is selected as the naming scheme for the class of reference data representing "Country".

The reference data to be loaded may originate from multiple data sources using different naming schemes for the same reference data. The data processing system of the embodiment supports the use of different naming schemes by allowing the user to define such different naming schemes before loading the data. On loading, if the used naming scheme is unknown, the data may be rejected or buffered to allow a new naming scheme (e.g. new name corresponding to an existing product or company entity, or new entity) to be added.

At step 101, measures are defined. This is done by entering a name for a new measure, and entering or selecting a unit (and/or type of unit, such as "length") to be associated with the measure. For example, a new measure may be "Cost of Manufacturing" which is associated with the unit "Pound Sterling". The measures include those associated with the raw data present in transaction records themselves; for example weight, cost, price, length, viscosity and so on. These are referred to as "stored" measures. They also include those derived from the data stored in the transaction records. These

comprise measures derived by stored predetermined unit conversion operations (such as centimetres to inches); those calculated by a formula from others (such as density from weight and volume); and those aggregated from others. These latter include measures derived by aggregation over time (such as volume per month aggregated from daily volumes or all sales volumes); and measures aggregated over another dimension. Some measures (e.g. temperature) cannot meaningfully be aggregated at all. For each such measure, the stored record includes association records indicating its place in a hierarchy (for example, "kilogram" as an instance of a unit of weight) and the formula for calculating it from other measures where necessary.

Similarly, at step 102, classes of transaction data are defined. A class of transaction represents a holding place for transaction data entries. A user may define a class of transaction by entering a desired name for that class, and by selecting a number of dimensions and measures from the previously defined classes of reference data and measures, respectively.

For example, to create a class of transaction data in accordance with the schema illustrated in Figure 4, the user would have to select the dimensions Delivery Date (box 42 in Figure 4), Delivery Point (box 43), Packed Product (box 44) and Sales Representative, as well as the measures Volume and Manufacturing Costs (box 41) and its associated units.

Having thus been input at the workstations 22, the metadata is stored in the Oracle™ database held within the storage device (e.g. large capacity disk device) of the server 21.

Loading Reference Data

At step 103, the reference data is loaded into the storage means of the server 21. Reference data to be loaded may, for example, consist of a list of Product Families. Such a list is provided, for example, in the form of a spreadsheet in Microsoft Excel (RTM)

In order to convert the list into the format required for storage in the reference data table, an Import File Definition (IFD) has to be defined by the user. The IFD may only be used for loading one class of reference data. For example, the reference data to be loaded may be a list of Product Families which are managed by a Brand Manager.

The IFD has to be defined by the user such that the input file for receiving the external data matches the source file format.

The user then also has to include into the IFD a definition of that association between the Product Families and the Brand Manager. This is done by first selecting the class of reference data for Product Family (representing the actual reference data to be loaded), and then by selecting an association of that class of reference data with the class of reference data for Brand Manager. The loading may then be initiated. The reference data is

stored, in the way discussed in connection with Figures 6 to 8b, in the Oracle™ database held within the storage device (e.g. large capacity disk device) of the server 21.

On loading of the reference data, the loaded data is verified against the definition of the selected classes of reference data and their associations as well as their defined naming schemes. If a selected class of reference data is associated with a parent class of reference data (i.e. a mandatory association), the user has to select the action to be taken by the data processing system if the loaded reference data corresponding to that parent class of reference data uses a naming scheme which is not defined in the data processing system.

The user may select one of three available actions, namely to reject just reference data elements which use an unknown name, to reject the entire batch of reference data, or to include a new definition in the data processing system such as to support the new naming scheme (i.e. name foe existing entity, or new entity) of the reference data to be loaded. In the latter case, a new record of reference data is created by the user using the code and the name as required by the reference data to be loaded.

In order to provide for the above, the user has to include into the IFD the measures which are required to be included, the units for each measure if they have been defined as variable, the classes of reference data to be included, the action to be taken if an element of reference data does not exist, and the action

to be taken on any associated reference data element according to the metadata definition to ensure complete integrity of the reference data.

The actions can be the creation of a new reference data element, the creation of a new parent if the new reference data requires such association according to the metadata definition, the modification of a parent reference data element in order to ensure that the metadata definition of time variant hierarchies or many to many relationships are obeyed, or the release of a reference data element if it is no longer relevant whilst retaining it so that historic information relating to that reference data element is retained.

The invalid reference data is stored in a holding area for subsequent correction by the user. The corrections can be made by searching for reference data already stored in the data processing system and selecting the correct data element, or by creating a new element directly from one of the workstations 22a-22c such as to render the reference data valid.

Accordingly, the data model used in the data processing system is adaptable on loading of external data such as to support the loading of data the format of which is unknown before loading.

If the selected class of reference data has any optional association with a parent class of reference data then the user may also select whether or not the reference data to be loaded contains any details for that parent class of reference data.

As set out in connection with Figures 8a and 8b, each object contained in the data processing system may be associated with a period of validity comprising a start date of validity and an end date of validity. The start date of validity is set on loading of the reference data. By default, the start date contained in each reference data element is defined as the date of loading. However, the start date may also be input at a workstation 22 by the user if a date different from that of loading is desired. The end date may be input by the user on loading, but is often not set on loading but subsequently, with a change in a business entity (e.g. on a reorganisation) on the date when an object becomes invalid, for example when an association ceases to be valid, because it has been deleted or replaced by another incompatible association.

If the association is hierarchical, the end date is set when a new parent business entity is defined. It is thereby guaranteed that there can only be one parent reference data element for a child reference data element at any time. Accordingly, loaded transaction data is referenced to the corresponding reference data only once.

Loading Transaction Data

Having initialised the system, at step 104, the transaction data is loaded into the data processing system. Although this is shown as a single step, in practice for a data warehouse, transaction data of different types will be

loaded periodically; some transactions will be loaded daily, some weekly, some monthly and so on.

This is realised by the user creating, for each type of transaction, a File Definition by selecting one of the classes of transactions defined previously, and then selecting from that class of transaction a sequence of one or more dimensions and one or more measures, in the order in which they occur in the fields of the records of transaction data received from the data sources 24a, 24b. The user may select units different from those associated with a selected measures.

Then, the transaction data is loaded into the storage means of the server 21 which embodies the data processing system of the embodiment, and stored therein in the format illustrated at Figure 5. If the transaction data before loading is in a format different to that of Figure 5, it is converted into this format on loading. In other words, all transaction data for a given transaction type is stored in the data processing system in the same standard format.

Invalid transaction data (transaction data not matching the metadata definitions, or including unknown names of reference data entities) is stored in a holding area for subsequent correction by the user. The corrections can be made by searching for transaction data already stored in the data processing system and selecting the correct data element, or by creating a new element directly from a user terminal such as to render the transaction data valid.

The transaction data to be loaded not only includes numeric values but also one or more codes representing the above explained naming scheme. From these codes, the data processing system identifies against which reference a transaction is measured and generates the pointers contained in a transaction data item as shown in Figure 5.

Accordingly, each stored transaction data item includes a number of fields holding numeric values (see fields 55 and 57 at Figure 5), a number of fields holding pointers to the associated elements of reference data (see fields 50 to 54 of Figure 5), and pointers to the units used (see fields 56 and 58 of Figure 5).

Display and Editing of the Model

Once the data processing system is initialised in the above described way, the user may display the stored data. In particular, the user may display the metadata (classes of reference data and their associations to one another). The user may also display the reference data elements classified under the different classes of reference data.

It is thus possible to view the business model comprising the structure of the organisation and its customers and suppliers, which is reflected by the classes of reference data and the associations between each other, and the actual reference data representing "instances" thereof. Also, it is possible to display the periods of validity of the associations between those instances.

This permits the viewing of how the underlying business organisation has changed over time.

Figures 17a and 17b illustrate a first view produced at a display of a workstation 22 under control of the data browsing program forming part of the control program of the server 21 and using a GUI. This provides a view corresponding to the "Explorer" program provided with Windows™. Successively lower layers of the hierarchies of reference data and metadata can be displayed, as shown in Figure 17b, to allow the user to see the definitions of classes of business entity, and the elements stored for each class.

Figure 18 illustrates a first view produced at a display of a workstation 22 under control of the GUI. This tool is a data structure browser, which shows, for each element of reference data or metadata, the layers of data hierarchically above and below that element. This enables the user quickly to grasp which reference data can be used as dimensions across which to analyse a given measure, or which measures can be analysed over a given dimension. The GUI is accordingly arranged to respond to the input devices of the workstation, to browse the stored metadata and reference data held within the server 21, and to generate the graphic display of Figures 17 or 18.

The data model may be adapted to represent such changes in the business organisation. For example, a brand manager may have taken over the management of another brand. To reflect such change, the association of

that brand manager with the brand name is adapted. As shown in Figure 21, this is done by creating a new association, with the date of the change as the start date of validity, while the existent association is retained, with the date of the change as the end date of validity.

It is important to note that despite the adaptation, the reference data element representing the brand manager's association with the brand name prior to the business re-organisation is retained in the data processing system so as to allow viewing of the reference data before and after the business re-organisation.

This is achieved by the data processing system utilising the period of validity information which is attached to each association so as to display the time variant reference data. The date as of which the data is to be analysed is compared with the periods of validity of each association, and those for which it lies within the period are utilised for analysis as discussed below.

Particular typical hierarchical structures

As an illustration of the manner in which the invention can be used, two typical hierarchies will briefly be illustrated. Firstly, the "product" hierarchy provides various ways of describing a given product. Metadata is provided which provides classes for saleable product and, hierarchically below that, product subgroup and product group.

Each reference data record which instantiates one of these classes may be linked with multiple differential textual names.

Products are also classified according to an alternative hierarchy of technical grade; for example, by bands of viscosity or weight. A given type of product (represented by a reference data item) may therefore be a member of several different product hierarchies.

Organisational elements are also typically provided with predetermined classes consisting of organisation; department; delivery point; individual and so on. Alternative hierarchies also provided may, for example, consist of geographical classes of entity such as region, country, district, town and so on. A given organisation of unit may therefore be a member of several hierarchies based on position in organisation, location and so on.

Variable Depth Classification

Figure 11 shows an illustration of a classification of products. Row 110 includes a hierarchical product classification. Row 110 represents the Classes of Business Entities, while rows 111 to 114 represent Business Entities ("instances"). Rows 111 to 114 illustrate products A to D and how these are classified. Products A to D are classified in different ways; for example products A and D have no "Product Sub Group" classification and product C has no "Product Sub Group" and no "Product Group" classification, while product B includes all available classifications.

Figure 12 illustrates how different classification structures may be used concurrently in the data processing system. A Class of Business Entity at one level can be linked with another Class of Business Entity at any other desired level. The levels correspond to the columns in Figure 11. In the shown example, Product Class 120 is associated only with Product Sub Class 121. Product Sub Class 121, however, is associated with both Product Group 122 and Saleable Product 123 (in accordance with product C of Figure 11). Similarly, Product Group 122 is associated with both Product Sub Group 124 and Saleable Product 125, and so on . Accordingly, variable depth hierarchies can be realised in the data model of the embodiment.

If a new product is to be included, the data model does not need to be adapted if the new product is classified differently. In contrast, the new product is simply incorporated in the existing hierarchy since the data model supports a variable depth classification of the new data. For example, if a product E (Saleable Product) was classified as a sub-class only of Product Class in Figure 9, then a direct association with Product Class would be created.

However, if the underlying business organisation changes, the hierarchy can be adapted to reflect such change. For example, if another level such as “Product Sub Sub Group” is to be included, this could be realised by creating and including a new Class of Business Entity without impacting the data stored in accordance with the previous hierarchy. The new level can then

optionally be used for classifying some part of the business entities. Thus, in this scheme, each reference data record for a business entity refers to (points to) others above and below it in the hierarchy of which it is part, and these also refer to correspondingly hierarchically arranged levels of classes of business entities in the meta data.

An alternative is to use so-called involutions. In this case, records for business entities are arranged in a hierarchy, but are not allocated hierarchically arranged different classes of business data within the meta data; instead, all are instances of the same class. For example a single meta data class of reference data for "Department" in a business organisation may be used for different instances at different levels, to provide a business classification.

Figure 13 illustrates how a variable depth hierarchy is represented by using involutions. Figure 13 shows different instances of a Class of Business Entity "Department" 131 at different hierarchical levels. The associations between the different hierarchical levels are defined by involutions as set out above. Accordingly, the "company" record 134 is linked as parent to the "distribution" and "sales" records 135 and 136, the latter likewise being linked to the "retail" and "commercial" records 132 and 133, "retail" 132 being linked to "general" 137 and "retail" 138, and "commercial" 133 to "government" 139, to map the structure of a given organisation. Each

indicated link is provided by an association record with a stored validity range, as discussed elsewhere.

Querying and Extraction of Data

The data processing system also allows a user to query the transaction data and to display the queried transactions. This is done by the user selecting one or more reference data elements (dimensions) and measures against which the selected dimensions are to be displayed. Thereby, the transaction data which is measured against the selected dimensions is retrieved.

More specifically, the data processing system allows a user to select and combine data from across multiple transaction datasets in order to generate a virtual hypercube for subsequent use by an analysis tool such as Microsoft Excel™. The different selected transaction datasets may represent a combination of transaction datasets for the same underlying class of transaction, the form of which, however, varies over a selected period of time as additional measures are captured or the dimensions against which the transaction measures are analysed vary in some way.

Also, the user may select transaction datasets from different underlying classes of transaction containing different measures, but which are analysed against one or more common dimensions.

Referring to Figure 22, the process comprises the steps of:

- Defining the date for analysis;

- Inputting the desired measures and dimensions across which they are analysed, together with any constraints on those dimensions (e.g. a date limit);
- Selecting the transaction records needed for the analysis; and
- Calculating and/or aggregating the data therefrom, where necessary, to match the dimension selected for analysis.

Figure 19 illustrates a view produced at a display of a workstation 22 under control of the GUI, to enable data extraction to be performed graphically.

Since all transaction data items are provided with a transaction date, and all associations between dimensions are provided with periods of validity, it is possible to display historic information reflecting transactions that have taken place at any desired date irrespective of changes in the underlying business organisation after the desired date. Specifically, as shown, this embodiment provides three choices for analysis of the transaction data:

- As of the date of the transaction - i.e. using the associations between business entities which were valid on the transaction date (this is the default);
- As of the current date – i.e. using the associations between business entities which are valid at the current date; or
- As of some specific, user-input, date.

Thus, it is possible to generate projections on the basis of the historic information to determine how a business would have developed had a re-organisation not taken place by selecting, as the chosen analysis date, a date prior to the reorganisation; or to project the current structure backwards in time as if it had always existed whilst past transactions were taking place.

Once the analysis date has been supplied, the selected associations (those having matching validity periods) define the business model which is to be applied to enable the data to be analysed. Thus, when a given measure is specified (for example, price of a certain product featuring in one or more specific transaction types) and a dimension against which it is to be analysed is supplied (for example, customer region), the data extraction process performed by the server 21 is arranged to read the stored reference data and metadata indicated by those associations, and to determine whether, and how, the analysis can be performed.

If all transaction records containing a reference to that product also contain a reference to the desired measure (price) and dimension (customer region) then selection of the records required for the analysis is simple. Likewise, if transaction records contain a reference to a dimension (e.g. "customer" or "customer delivery point") hierarchically below that chosen, extraction is possible since such records can be mapped unambiguously to the desired dimension using the stored associations.

Where the business structure has changed, for example to cease to record a given reference data item such as "customer region" for all or some transactions, then only those transaction records which have dates for which the association with the desired dimension are valid can be analysed by that dimension.

The query interface only requires the user to specify the data (measures) they wish to see, i.e. to analyse against dimensions. The data processing system determines what sources (transaction datasets) are available for the data that may be used to satisfy the query. Several different transaction data may be available as alternatives, where, for example, both daily and monthly sales or inventory figures are archived. If the analysis requires only a monthly breakdown in the time dimension, it is more economical to refer only to the monthly transaction records.

Accordingly, in general, the data processing system of the embodiment is arranged to determine which of plural different sets of transaction records including the same data is closest in the hierarchies of dimensions and measures to those sought for analysis.

The data processing system of the embodiment is also arranged to determine how to formulate a set of underlying queries to extract and manipulate the necessary data in the required form. The user may also include constraints to limit the data to be analysed and/or presented (for

example, to a certain date range, to a certain range of products, or some other limitation affecting one or more dimensions).

Where possible alternative sources of transaction data exist, the data processing system evaluates the possible options in order to select the set of sources which will, (where necessary within a predetermined margin of uncertainty), most cheaply (in terms of processing overhead) satisfy the requirement. In this way, for example, the data processing system may automatically make use of transaction datasets that have been pre-summarised in one or more dimensions to reduce the volume of data to be processed.

Specifically, for each possible set of transactions records, the processor checks the start and end dates of the records available to see whether they correspond to the range of data requested. Next, the processor determines whether all requested measures and dimensions can be derived from each class of transaction records. If only a single class corresponds to the data constraints, dimensions and measures required then that is selected.

If more than one class permits the required measures to be derived over the required dimensions, or if some can approximate the required data, then each transaction data set is allocated a "score" indicating how closely the data available matches that sought (how many levels of hierarchy from that sought it can reach) and the number of calculations required to calculate the desired measures and dimensions from those available.

If several classes of transaction data have the same score, then the smallest set (the one with the least number of records) is selected.

If the data cannot be provided from a single transaction record set over the whole period sought, but is available over part of the period sought, then the processor is arranged to re-analyse the remainder of the period, to determine whether other transaction data sets can provide the data over the remainder of the period.

Data from different transaction types can be jointly utilised by the data processing system of the embodiment to generate an analysis, since it shares at least some commonly defined business entities. However, data from different transactions may not uniformly reference the same levels of the dimensional hierarchies – some transactions may record, for example, customer delivery point of a sale whereas some only record the customer.

In combining data from multiple sources, the data processing system of the embodiment will, where necessary, automatically aggregate data up common dimensions in order to arrive at shared reference data elements – i.e. to reach the lowest reference data element in the hierarchy which is accessible from all transaction data to be used in the analysis (the customer, in the above example, since analysis by delivery point is not possible for all transactions).

Thus, in performing an analysis by customer, records for all transactions referenced to delivery points which are associated with that customer at the

analysis date are selected, and the measures therefrom are cumulated to form a total for that customer.

Measures may either be taken directly from transaction datasets (aggregated up the dimensional hierarchies as appropriate) or may be derived by calculation. Measures may be "derived measures" calculated from a number of underlying measures by applying a formula, for example to calculate a 'cost per litre' measure from a 'cost' measure and a 'volume' measure. Data defining the necessary formula is stored in the reference data element defining the derived measure. The underlying measures may be stored measures (i.e. those stored in transaction data) or may themselves be derived measures; they may also be drawn from more than one transaction set.

Also, measures may be derived by aggregation against one or more reference data elements; for example, a measure for sales of a particular product or sales over a particular period of time. The measures so derived may themselves be used in further calculations. For example, they may be used to derive a figure for the percentage increase of sales for the current year to date over the corresponding period in the previous year.

Measures denominated in currencies may be converted to one or more specified currencies. The data processing system provides support for multiple sets of exchange rates. For example, exchange rates may be drawn from different sources or for differing periods of time (daily, monthly, quarterly, yearly, etc.). The user may specify that the exchange rates used for

converting the measures are the rates current at the time of the transaction (in order to account for exchange rate fluctuations), or the rates current at some particular point in time (in order to allow comparisons over time with exchange rate fluctuations masked out).

Thus, it will be seen that on the user specifying the date for an analysis, and the desired measures and dimensions for the analysis, the data processing system of the embodiment is able to utilise the above-described stored data structures to determine possible sources of transaction data for the analysis; to select a source or sources which most closely match the desired analysis (or, where a choice exists, minimises the amount of calculation required to aggregate data); to aggregate the selected transaction data to match the desired level of analysis; and to output a file of data including, for each element of reference data in the selected dimension(s), a value for each selected measure. The file may be transmitted to a workstation 22 as an Excel ™ workbook, or a binary file for processing in another format, or may be stored on the server 21 itself for future use.

On retrieval of data from the data processing system, the user may display historic information on the basis of different "types" of time. The data processing system supports five different types of time grouped in three different classes.

The first class is the "Specific" time class. The "Specific" time class covers two types of time periods, namely fixed periods (e.g. year, quarter,

month, day), and current periods based on the current system time (e.g. today, this month, yesterday).

The second class is the "Relative" time class. It covers two types of time periods, namely relative period (e.g. year to date), and corresponding periods (e.g. previous year to date).

The third class is the "Typical" time class which covers typical periods of time repeat, such as Tuesday, Christmas day, etc.

Thus, the data processing system provides a flexible way to represent time and allows the implementation of any calendar such as for example the Chinese calendar or the Islamic calendar. This enables the user to summarise data based on groupings of time against a required calendar which is not restricted to the western Gregorian calendar.

Example of editing Business Model

An application of the data processing system for the storage of time-variant business information is now described in connection with Figures 14 to 16.

As set out above, all transactions stored in the data processing system comprise a date of transaction. In addition, all associations between Business Entities as well as associations between measures and units are associated with a period of validity. This allows a proper tracking of changing conditions of a business.

Figures 14 to 16 illustrate how the data model can handle changing business requirements. The shown example refers to an oil products distribution company, which has two divisions, each with a set of distribution managers, who are in turn responsible for customers. Each of the rows 140 to 143 shown in Figures 14 to 16 corresponds to a Class of Business Entity, representing the division (row 140), the distribution managers (row 141), the delivery points (row 142) and the customers (row 143).

Figure 14 shows the business situation at a first date. The distribution managers Brice 144, Harcroft 145 and Smith 146 each are responsible for one or more of delivery points, and each of the delivery point is associated with one or more customers. However, at some time after the first date, the business structure is reorganised, and the distribution manager Brice 144 is moved to the Retail Division 147 to meet an increased demand from one of the customers, Abott's Autos 148. The restructured business is shown in Figure 15. Subsequent to this business reorganisation Abott's Autos 148 takes over two other customers, Auto Stop 149 and Raydon Wharf 150. This is shown in Figure 16.

In a traditional data processing system, such external business reorganisation would be difficult, if not impossible, to deal with. As a consequence, the data warehouse would be likely to lose historic information. By contrast, in the data processing system of the embodiment, the data model can be adapted to the changed requirements as explained above. However,

since the transactions as well as the associations between Classes of Business Entities are provided with time information, no data is lost on adaptation of the data model. Rather, it is still possible after the adaptation to retrieve and display data from before the adaptation. This makes it possible, for example, to compare data collected before and after a business re-organisation. Accordingly, the data processing system can be used to evaluate the consequence of a business reorganisation.

Summary

It will be seen that the above-described embodiment illustrates the following features. It allows volumes of transaction data to be input and stored. The transaction data may represent multiple different types of transactions. The business entities involved in the transactions (products, companies and personnel) are defined in separately stored reference data, structured in accordance with stored metadata.

The relationships between the business entities and the metadata classes to which they belong are related by stored association records. Thus, different transaction records storing different levels of granularity of information on such business entities can be aggregated using such stored association records.

Each such association record has a period of validity, and each transaction record has date data. When the relationship between business entities changes, and/or a business entity is added or removed, existing

association records are kept, but their periods of validity may be amended, and new association records may be added. Thus, data defining the business model when each transaction took place is available for use in analysis.

On extraction of information, an analysis date or dates can be selected, and used to select the desired business model (defined by the association records valid for that date) to analyse the transaction data.

Use of metadata as described enables transaction data records to be input using a non-specific interface usable by non-programming staff, whilst providing the possibility of checking the validity of the input transaction data.

These and the other above-described advantages of the embodiment can be used separately of each other to provide their respective advantages in isolation if so preferred.

It will be clear that, whilst it is suitable for such use, the data processing system of the invention is not limited to a use in the field of business information management. Rather, the data processing system can be used in various other fields as well. For example, it can be used for monitoring chemical processes. Chemical substances could form the reference data, while classes of chemical substances could form the classes of reference data. The transaction data could be formed by the various parameters measured during a chemical process.

It should be noted that the present invention is not limited to the above described embodiment. It is envisaged that various modifications and

variations to the above described embodiment could be made without falling outside the scope of the present invention as determined from the claims.

CLAIMS:

1. A data processing system comprising a data storage device and a processor programmed to read data from, and write data to, said storage device, in which said storage device stores:

a) multiple operation records each storing data relating to one or more historical operation involving at least one entity, each said operation record comprising data recording the operation, and data defining a date associated with the operation; and

b) multiple entity records storing data indicating relationships between said entities, and each said relationship being associated with a historical period of validity.

2. The system of claim 1, wherein the processor is programmed to extract output data from a subset of said operation records, and to output said output data.

3. The system of claim 2, wherein the processor is programmed to select said subset by the steps of:

inputting instructions defining one or more selected entities for which said output data relates; and

selecting said subset based on both the dates stored in said operation records and the historical periods of validity associated with the selected entities.

4. The system of claim 3, wherein the processor is programmed to select said subset to represent by the steps of:

inputting an analysis date;

for the selected entities, selecting the entity relationships which have associated historical periods of validity within which said analysis date lies; and

selecting said subset using those selected entity relationships.

5. The system of claim 4, wherein the processor is programmed to offer the current date as a date option, to permit analysis of operation records anterior to that date as if the current relationship between entities had previously existed.

6. The system of claim 4 or claim 5, wherein the processor is programmed to offer an anterior date as a date option, to permit analysis of operation records posterior to that date as if a historical relationship between entities still persisted.

7. The system of any of claims 3 to 6, wherein the processor is programmed to analyse each operation record in accordance with the relationships between entities which have associated historical periods of validity within which the date of that operation record lies.

8. The system of claim 1, wherein the processor is programmed to input a change from an existing said relationship between entities to a new said relationship.

9. The system of claim 8, wherein the processor is programmed, on such a change, to store an end date for the period of validity of the existing relationship; to create a record of the new relationship, and to store a start date therefor.

10. The system of claim 1, wherein the entity records comprise:
an entity record for each entity; and

an association record for each past or present relationship between a pair of said entities;

each said entity record containing data representing its historical period of validity.

11. The system of any preceding claim, wherein the entity records comprise a hierarchical structure, in which at least a first entity record relates to a specific entity, and a second to a more generic entity encompassing said specific entity, said entity records including link data linking said first and second entity records whereby to allow said processor to traverse said hierarchy.

12. The system of claim 11, wherein the entity records represent first and second successive levels of hierarchy of an organisation.

13. The system of claim 11, wherein the entity records represent first and second successive levels of hierarchy of a product family.

14. The system of claim 11 when dependant upon claim 3, wherein said processor is programmed to:

input a historical analysis period; and

determine, for said operation records within said period, if said operation records relate to said selected entities throughout the whole of said period.

15. The system of claim 14, wherein, if said operation records do not span the whole of said period, for each selected said entity to which the operation

records relate, the processor is programmed to determine, from said entity records, a hierarchically higher entity and to repeat said determination and, in the event that said operation records relate to said hierarchically higher entity throughout the whole of said period, to use said hierarchically higher entity instead of said selected entity in selecting said subset of operation records.

16. The system of any preceding claim in which said storage means contains multiple sets of said operation records, each said set comprising multiple said operation records, said sets relating to different classes of operations and said records within each set relating to different instances of the same type of operation.

17. The system of claim 16, in which each said operation record contains at least one variable data field storing a value of a measure from a range of possible said values for said measure.

18. The system of claim 16 or claim 17, in which said storage means further contains:

c) meta data comprising multiple operation definition records, each defining the format of records of a respective said set of operation records.

19. The system of claim 18 when dependant upon claim 17, in which each operation definition record indicates the units of said measure.

20. The system of claim 16 or claim 17, in which said storage means further contains:

c) meta data comprising multiple unit definition records, defining the relationship between different said units.

21. The system of claim 17, wherein the processor is programmed to:

input at least one measure derivable from said operation records, to be analysed;

determine, for each said set of operation records, whether said measure can be derived therefrom; and,

where said measure could be derived from alternative said sets, select one of said sets.

22. The system of claim 21, wherein said selection is based on the relative sizes of said sets.

23. The system of claim 21 or claim 22, wherein said selection is based on the relative difficulty of deriving said measure from the data stored in the variable data fields of each of said sets.

24. The system of claim 17, wherein the processor is programmed to:

input at least one measure derivable from said operation records, to be analysed;

determine, for each said set of operation records, whether said measure can be derived therefrom; and,

where necessary, derive said measure from a combination of a first value from a variable data field of a record of a first set of operation records, and a second first value from a variable data field of a record of a second set of operation records.

25. The system of claim 17, wherein the processor is programmed to:
input at least one measure derivable from said operation records, to be analysed;

determine, for each said set of operation records, whether said measure can be derived therefrom; and,

where necessary, derive said measure from an aggregation of first values from respective variable data fields of a plurality of records of a first set of operation records, having dates spanning a predetermined input time interval.

26. The system of claim 1, wherein said operation records relate to respective transactions between said entities.

27. The system of claim 26, wherein said transactions are sales, inventory, or purchase transactions.

28. The system of any preceding claim, wherein said processor is programmed to load one or more new said operation records into said storage device.

29. The system of claim 28 when dependant upon claim 18, in which said processor is programmed to determine whether said new operation records comply with said meta data.

30. The system of claim 18, in which said processor is programmed to input said meta data.

31. A data processing system, comprising:

processing means for generating a data model in accordance with a data structure, the data model being adaptable to represent a change in the data structure; and

storage means for storing the data in accordance with the generated data model.

32. The data processing system of claim 31, wherein the stored data comprises information representative of the time of change in the data structure.

33. The data processing system of claim 31 or 32, wherein the stored data comprises:

transaction data representative of one or more measures which are determined relative to one or more references;

reference data representative of said one or more references; and

metadata descriptive of the transaction data and the reference data.

34. The data processing system of claim 33, wherein the metadata defines hierarchical associations between classes of the reference data.

35. The data processing system of claim 33 or 34, wherein the stored data comprises a number of elements of reference data, each element of reference data comprising information which defines an association with one or more other elements of reference data.

36. The data processing system of claim 35, wherein each element of reference data further comprises information representative of a first period of validity of a defined association.

37. The data processing system of claim 36, wherein the information representative of the first period of validity comprises a start date of validity and an end date of validity.

38. The data processing system of any of claims 33 to 37, wherein the one or more measures each are associated with one or more units.

39. The data processing system of claim 38, wherein the associations between the one or more measures and the one or more units are associated with a second period of validity.

40. The data processing system of claim 39, wherein the second period of validity comprises a start date of validity and an end date of validity.

41. The data processing system of any of claims 33 to 40, wherein the stored data comprises a number of items of transaction data, each item of transaction data being associated with a date of transaction.

42. The data processing system of any of claims 33 to 40, wherein the metadata defines associations between classes of reference data and the one or

more measures, the associations between the classes of reference data and the one or more measures being representative of classes of transaction data.

43. The data processing system of any preceding claim, further comprising:

first interface means for receiving data of any structure from a data source for storage in the data processing system.

44. The data processing system of any preceding claim, further comprising:

second interface means for outputting data from the storage means in a required format.

45. A data processing system, comprising:

processing means for generating a data model representative of data of a first structure, and for adapting the data model to represent also data of a second structure; and

storage means for storing data in accordance with the data model.

46. The data processing system of claim 45, wherein the stored data includes information representative of the time of adaptation of the data model.

47. A data storage device storing a data structure comprising:

a) multiple operation records each storing data relating to one or more historical operation involving at least one entity, each said operation record comprising data recording the operation, and data defining a date associated with the operation; and

b) multiple entity records storing data indicating relationships between said entities, and each said relationship being associated with a historical period of validity.

48. A data processing system comprising a data storage device and a processor programmed to read data from, and write data to, said storage device, in which said storage device stores multiple operation records each storing data relating to one or more historical operation involving at least one entity; and multiple entity records storing data indicating relationships between said entities, wherein the entity records comprise a hierarchical structure, in which at least a first entity record relates to a specific entity, and a second to a more generic entity encompassing said specific entity, said entity records including link data linking said first and second entity records whereby to allow said processor to traverse said hierarchy, said processor being arranged to generate output data by inputting instructions defining one or more selected entity dimensions across which said output data is to be distributed.

49. The system of claim 48, wherein, if all required said operation records do not relate to entities of the dimension to which the operation records relate, the processor is programmed to determine, from said entity records, a hierarchically higher level entity dimension and to repeat said determination and, in the event that all required said operation records relate to said hierarchically higher level, to use said hierarchically higher entity instead of said selected entity in selecting said subset of operation records.

50. The system of claim 48, wherein the processor is programmed to: input at least one measure derivable from said operation records, to be analysed; and determine, for each said set of operation records, whether said measure can be derived therefrom; and, where said measure could be derived from alternative said sets, select one of said sets.

Abstract

A data processing system is provided for storing and managing multiple data entries. The data processing system employs a data structure which allows the storage and management of a vast number of interrelated data entries the interrelations of which change over time. The data structure reflects such changing interrelations over time and allows the querying and extracting of data entries on the basis of their interrelations as they were or are defined at any desired point in time.

(Fig. 2)

This Page Blank (uspto

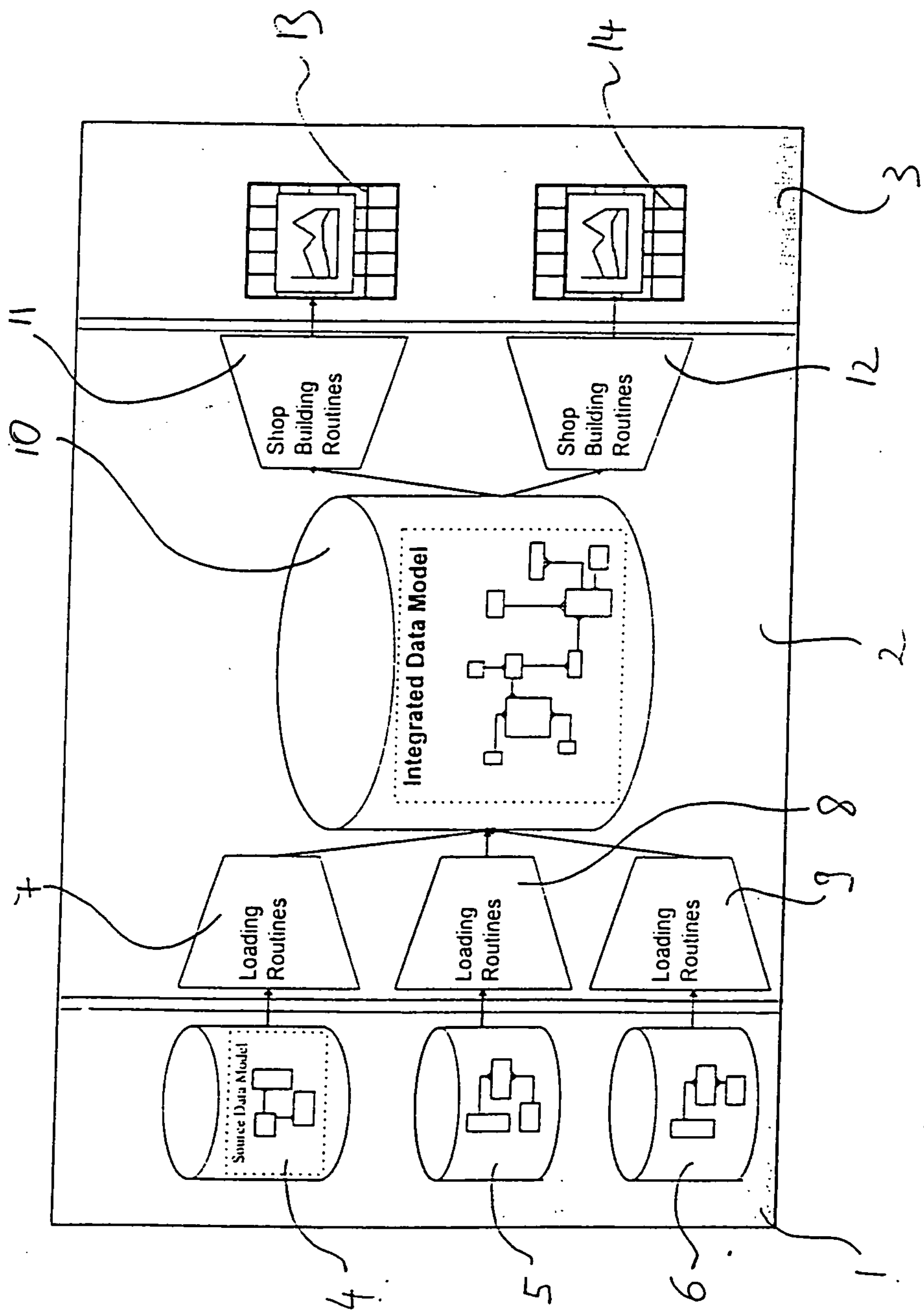

Source Data Model
Loading Routines
Loading Routines
Loading Routines
Integrated Data Model
Shop Building Routines
Shop Building Routines
1
2
3
4
5
6
7
8
9
10
11
12
13
14

Fig. 1

This Page Blank (uspto)

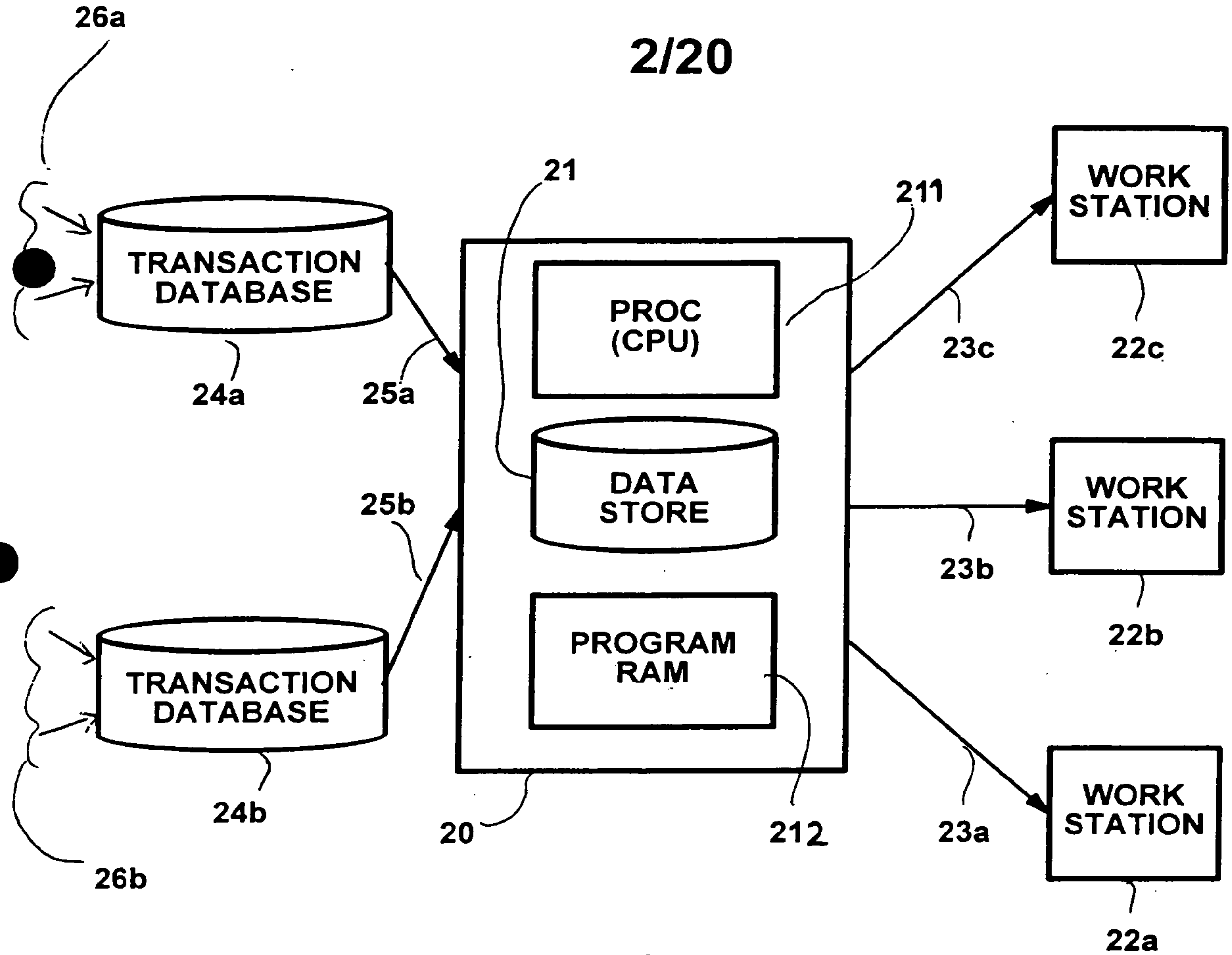

26a
TRANSACTION DATABASE
24a
25a
21
PROC (CPU)
211
DATA STORE
PROGRAM RAM
WORK STATION
23c
22c
WORK STATION
23b
22b
25b
TRANSACTION DATABASE
24b
26b
20
212
23a
WORK STATION
22a

FIG. 2

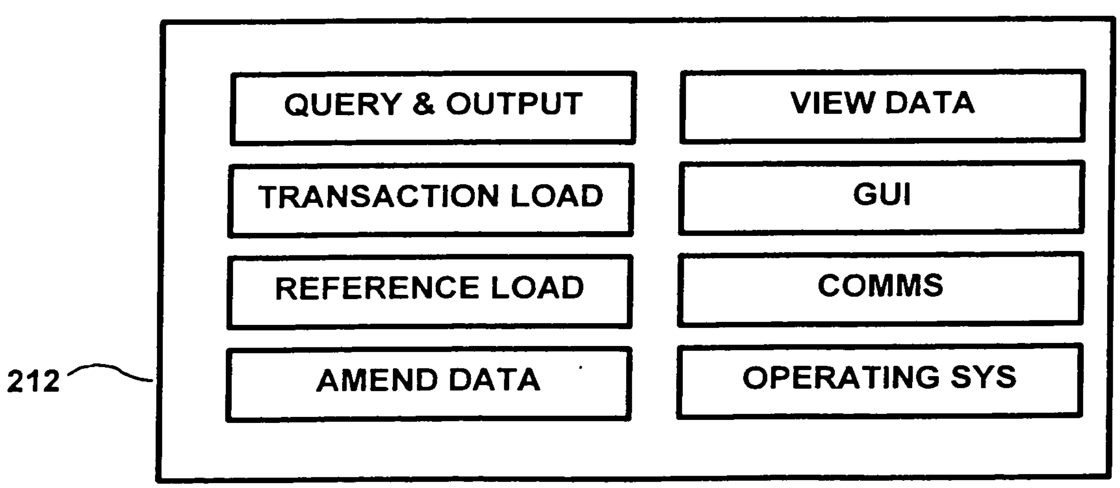

QUERY & OUTPUT
VIEW DATA
TRANSACTION LOAD
GUI
REFERENCE LOAD
COMMS
AMEND DATA
OPERATING SYS
212

FIG. 20

This Page Blank (uspto)

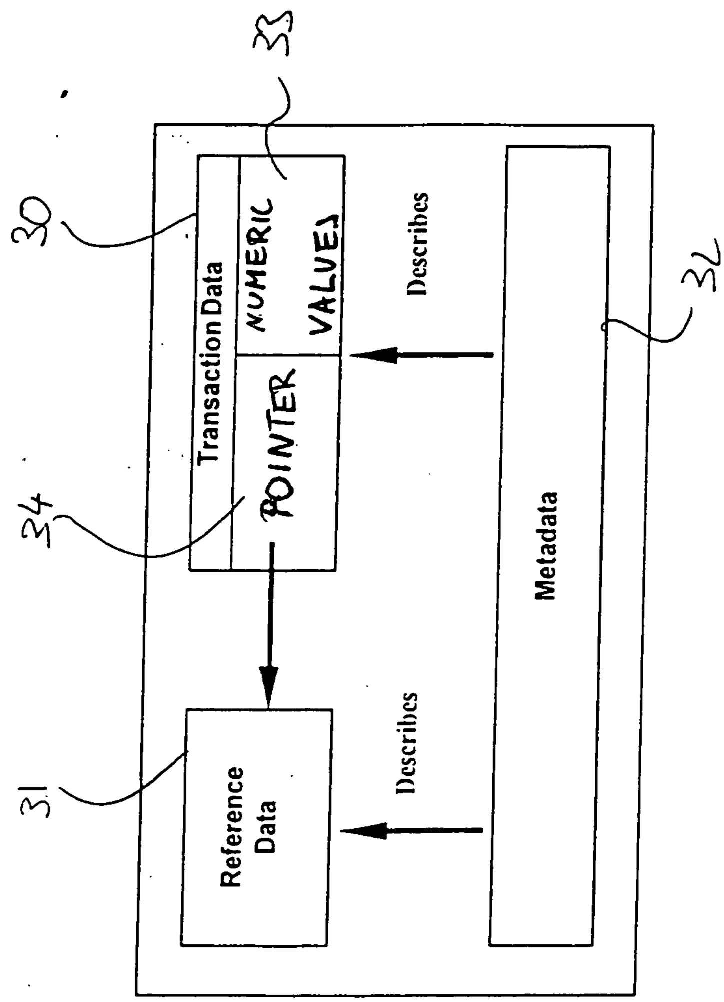

30
33
Transaction Data
NUMERIC VALUES
POINTER
34
Describes
32
Metadata
Describes
Reference Data
31

Fig. 3

This Page Blank (uspto)

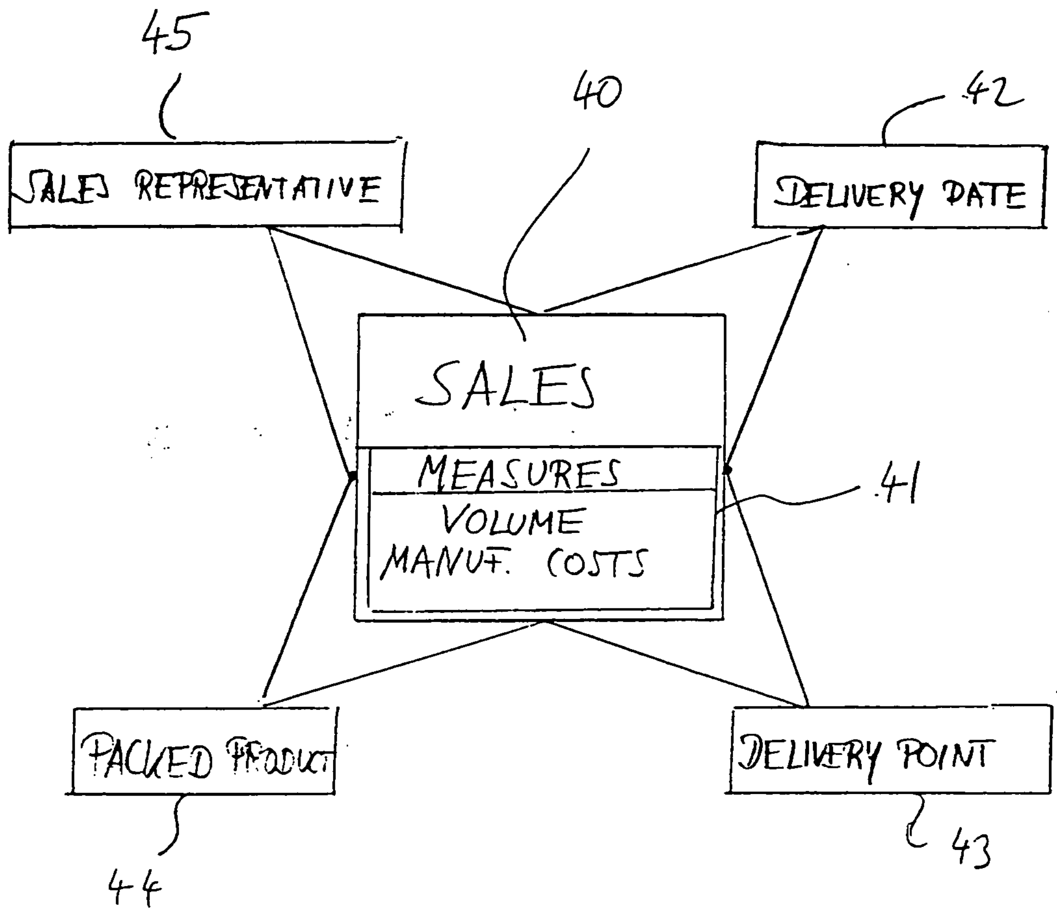

45
40
42
SALES REPRESENTATIVE
DELIVERY DATE
SALES
MEASURES
VOLUME
MANUF. COSTS
41
PACKED PRODUCT
DELIVERY POINT
44
43

Fig. 4

This Page Blank (uspto)

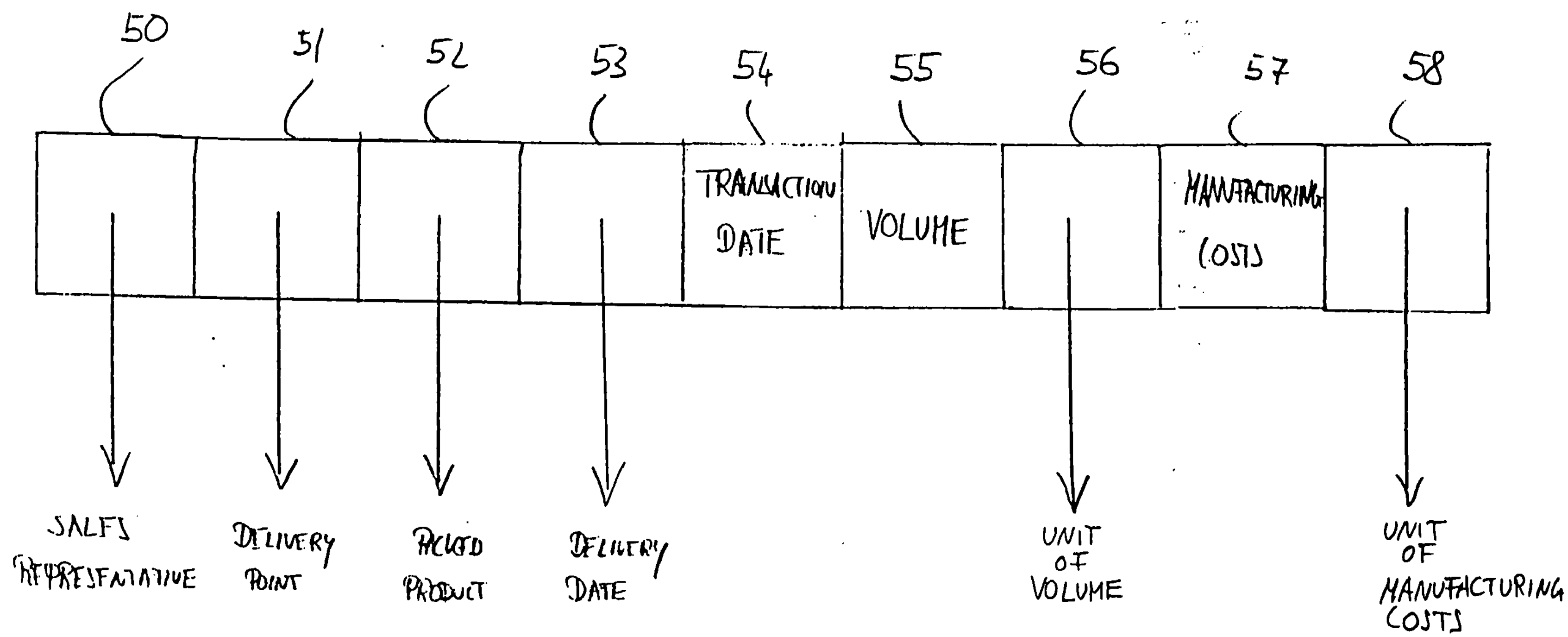

50
51
52
53
54
55
56
57
58
TRANSACTION DATE
VOLUME
MANUFACTURING COSTS
SALES REPRESENTATIVE
DELIVERY POINT
PACKED PRODUCT
DELIVERY DATE
UNIT OF VOLUME
UNIT OF MANUFACTURING COSTS

Fig. 5

This Page Blank (uspto)

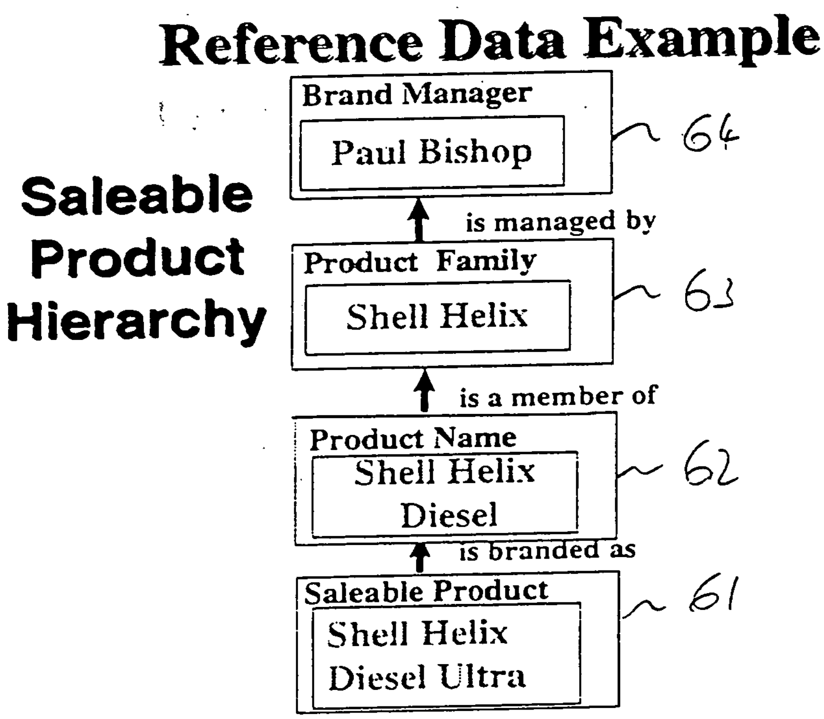

Reference Data Example
Saleable Product Hierarchy
Brand Manager
Paul Bishop
64
is managed by
Product Family
Shell Helix
63
is a member of
Product Name
Shell Helix Diesel
62
is branded as
Saleable Product
Shell Helix Diesel Ultra
61

Fig. 6

This Page Blank (uspto)

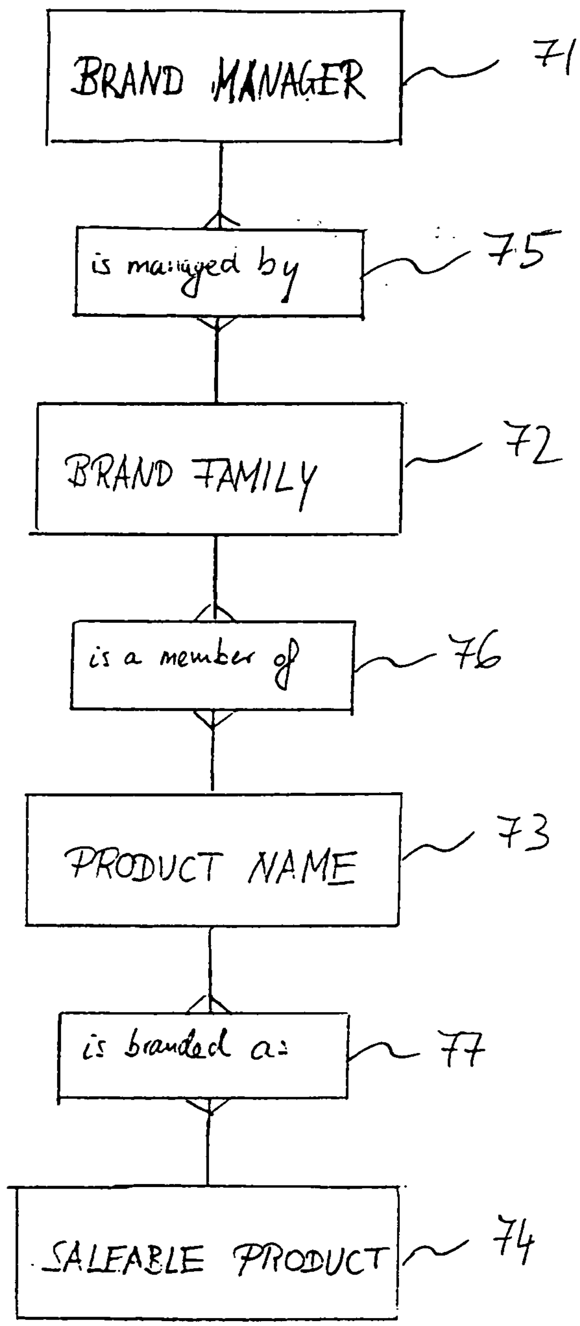

BRAND MANAGER
71
is managed by
75
BRAND FAMILY
72
is a member of
76
PRODUCT NAME
73
is branded as
77
SALEABLE PRODUCT
74

Fig. 7

This Page Blank (uspto)

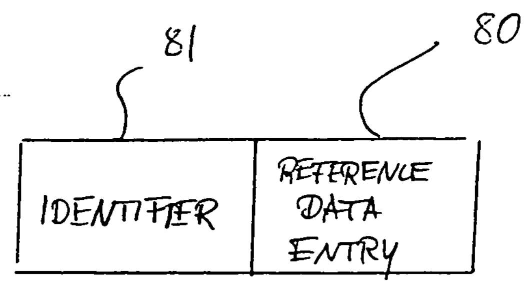

81
80
IDENTIFIER
REFERENCE DATA ENTRY

Fig. 8a

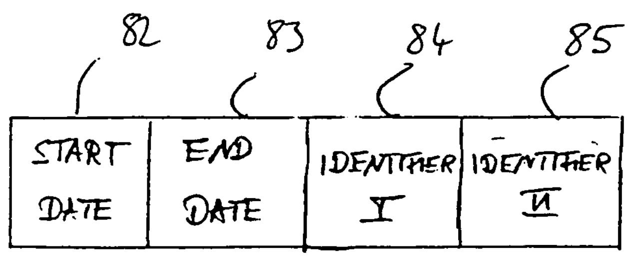

82
83
84
85
START DATE
END DATE
IDENTIFIER I
IDENTIFIER II

Fig. 8b

This Page Blank (uspto)

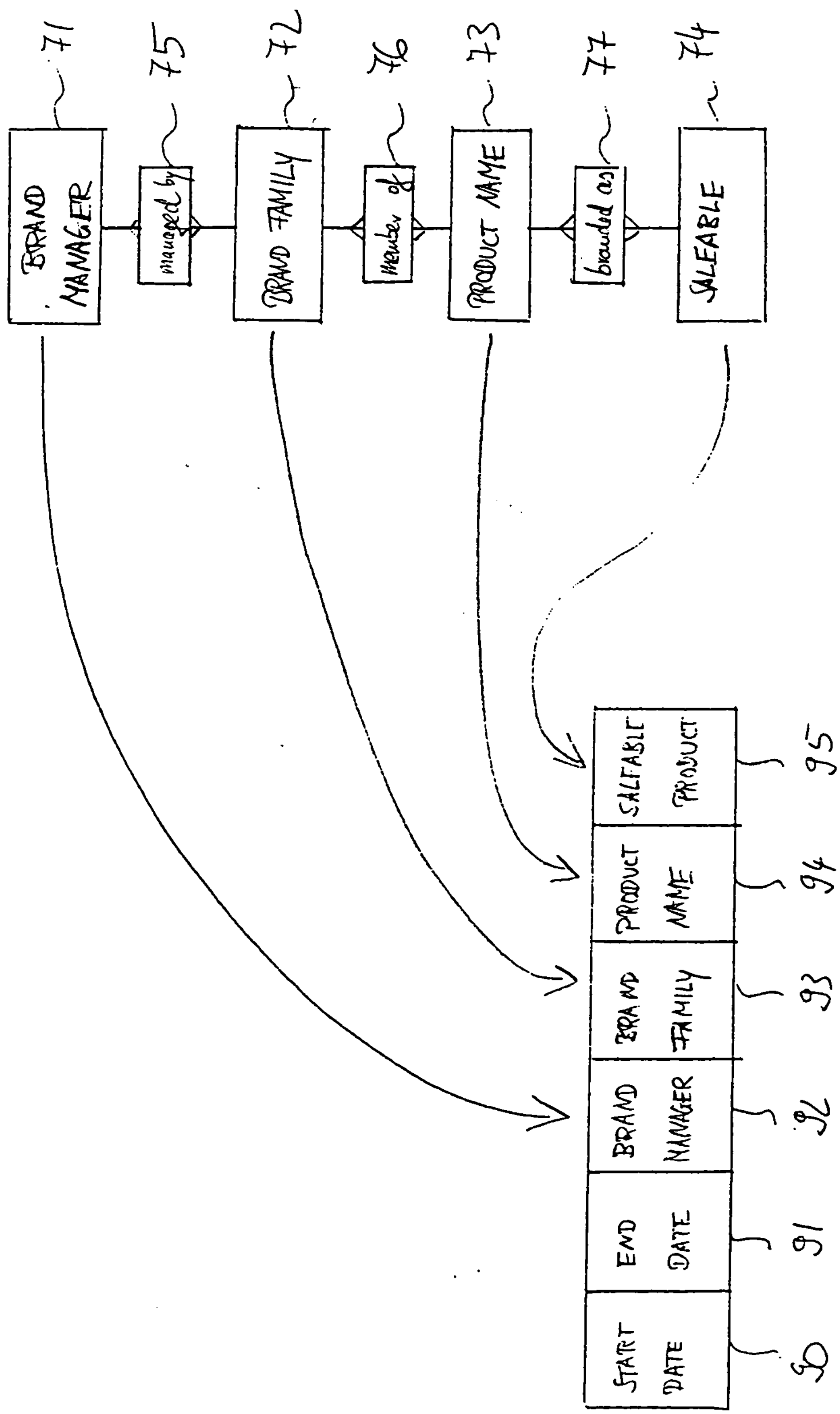
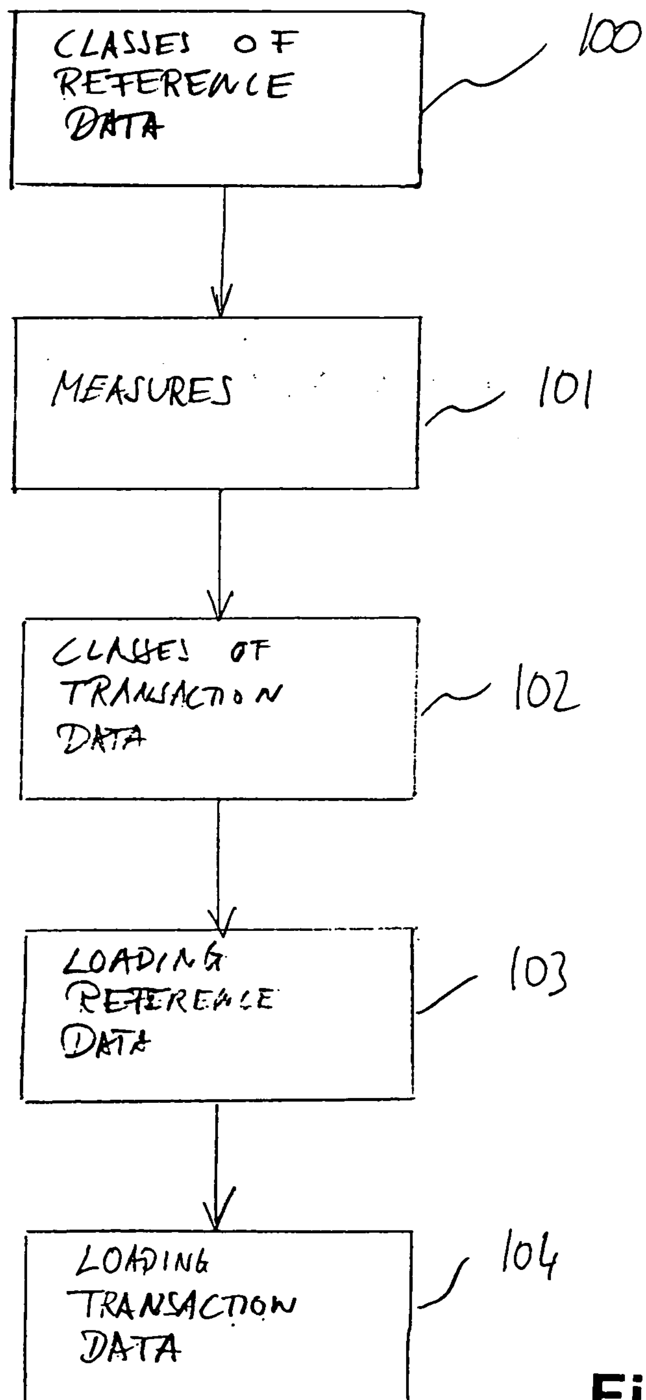

BRAND MANAGER
managed by
BRAND FAMILY
member of
PRODUCT NAME
branded as
SALEABLE
71
75
72
76
73
77
74
START DATE
END DATE
BRAND MANAGER
BRAND FAMILY
PRODUCT NAME
SALEABLE PRODUCT
90
91
92
93
94
95

Fig. 9

This Page Blank (uspto)



CLASSES OF REFERENCE DATA
100
MEASURES
101
CLASSES OF TRANSACTION DATA
102
LOADING REFERENCE DATA
103
LOADING TRANSACTION DATA
104

Fig. 10

This Page Blank (uspto)

Saleable Product	Product Sub Group	Product Group	Product Sub Class	Product Class	
					111
Product A	~	Trans-mission oils	Transport	Lubricant	112
Product B	Automatic Transmission Fluid	Trans-mission oils	Transport	Lubricant	113
Product C	~	~	Transport	Lubricant	114
Product D	~	Engine oils	Transport	Lubricant	115

Fig. 11

This Page Blank (uspto)

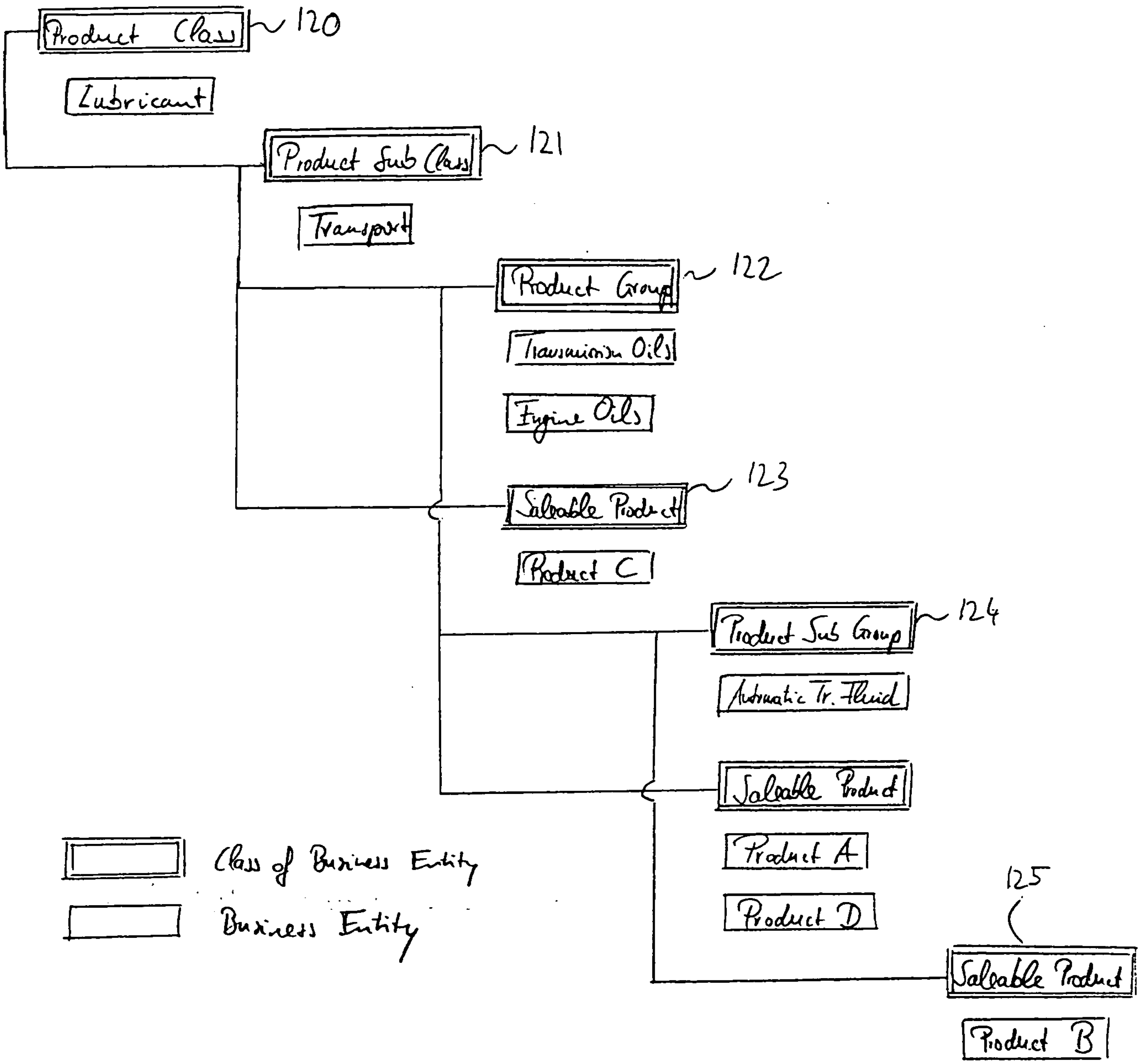

Product Class
120
Lubricant
Product Sub Class
121
Transport
Product Group
122
Transmission Oils
Engine Oils
Saleable Product
123
Product C
Product Sub Group
124
Automatic Tr. Fluid
Saleable Product
Product A
Product D
125
Saleable Product
Product B
Class of Business Entity
Business Entity

Fig. 12

This Page Blank (uspto)

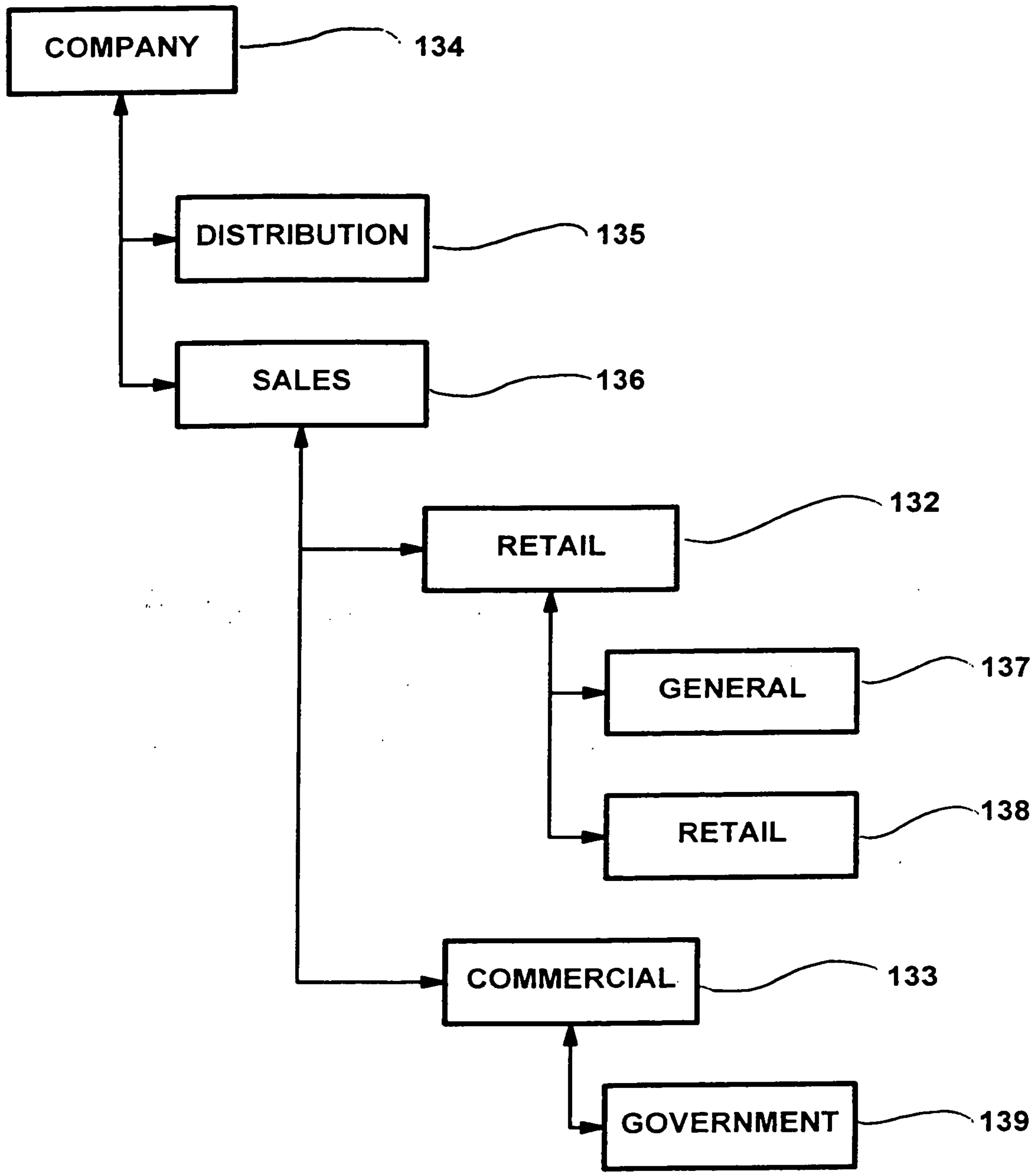

COMPANY
134
DISTRIBUTION
135
SALES
136
RETAIL
132
GENERAL
137
RETAIL
138
COMMERCIAL
133
GOVERNMENT
139

FIG. 13

This Page Blank (uspto)

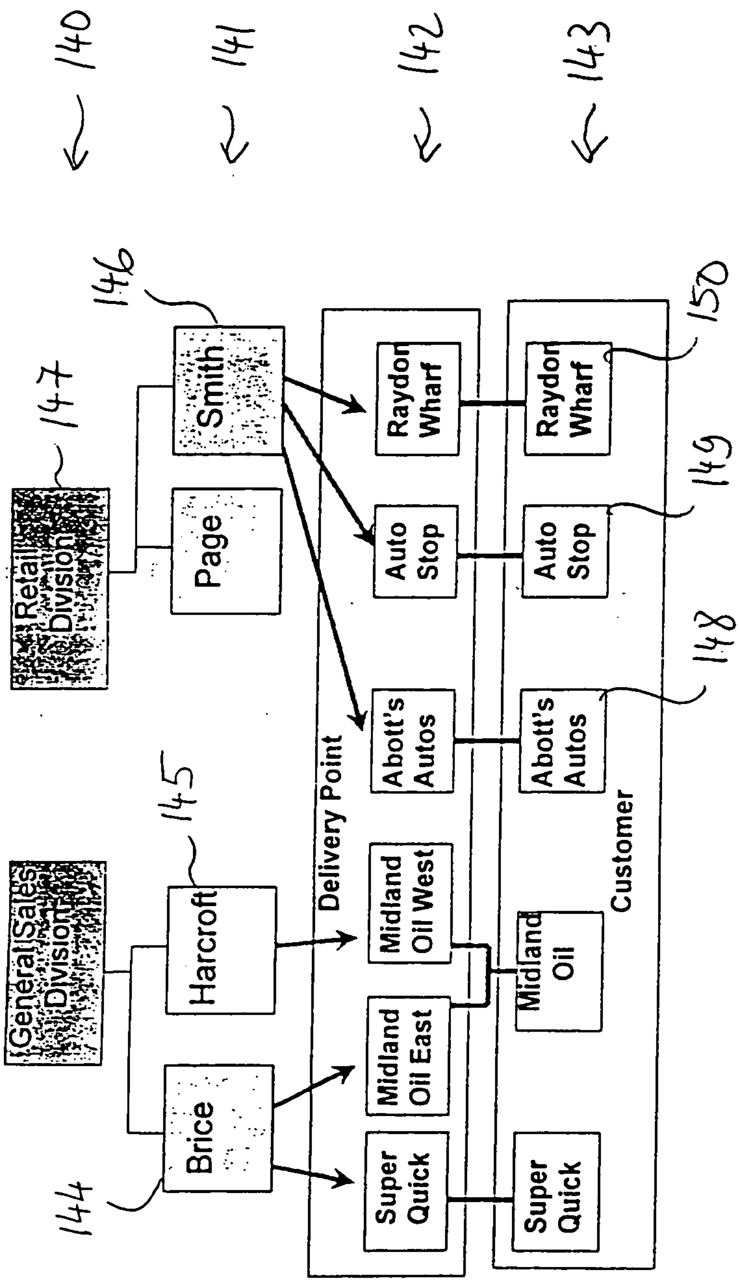
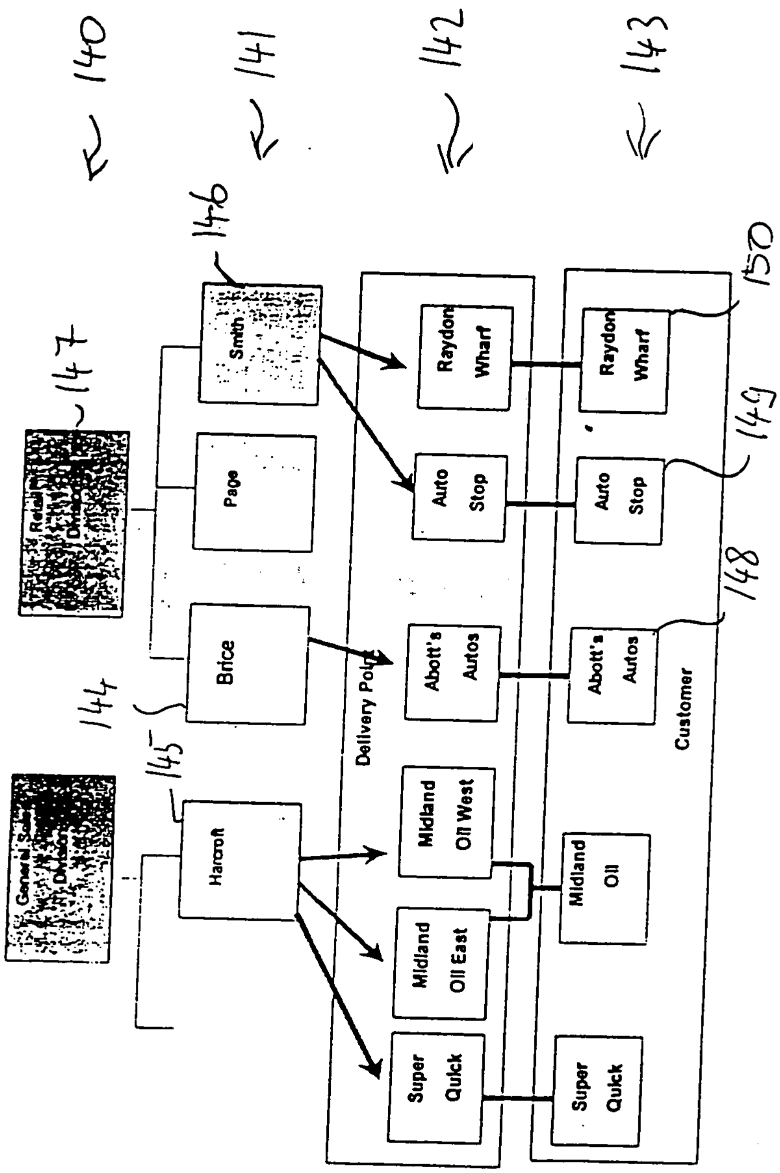

140
141
142
143
147
Retail Division
146
Page
Smith
General Sales Division
145
Harcroft
Brice
144
Delivery Point
Super Quick
Midland Oil East
Midland Oil West
Abott's Autos
Auto Stop
Raydon Wharf
Customer
Super Quick
Midland Oil
Abott's Autos
Auto Stop
Raydon Wharf
148
149
150

Fig. 14

This Page Blank (uspto)



140
141
142
143
147
146
150
149
148
144
145
Smith
Page
Brice
Harcroft
Raydon Wharf
Auto Stop
Abott's Autos
Midland Oil West
Midland Oil East
Super Quick
Delivery Point
Raydon Wharf
Auto Stop
Abott's Autos
Midland Oil
Super Quick
Customer

Fig. 15

This Page Blank (uspto)

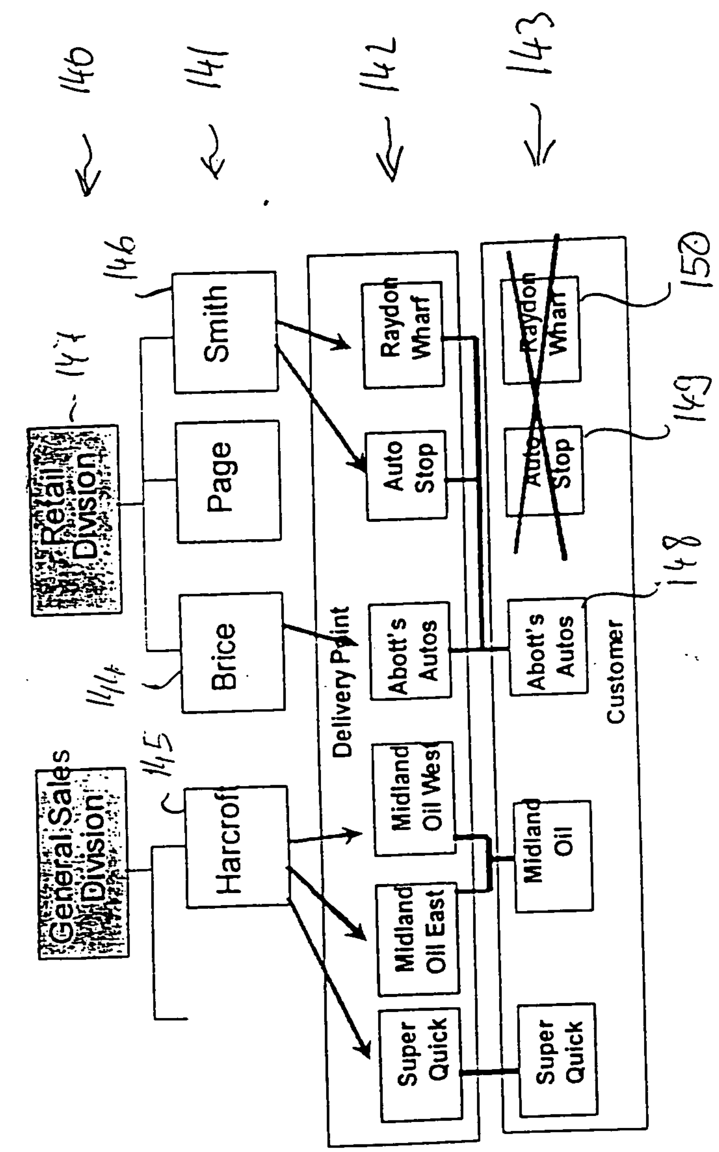

140
141
142
143
General Sales Division
Retail Division
147
145
Harcroft
144
Brice
Page
146
Smith
Delivery Point
Super Quick
Midland Oil East
Midland Oil West
Abott's Autos
Auto Stop
Raydon Wharf
Customer
Super Quick
Midland Oil
Abott's Autos
148
Auto Stop
149
Raydon Wharf
150

Fig. 16

This Page Blank (uspto)

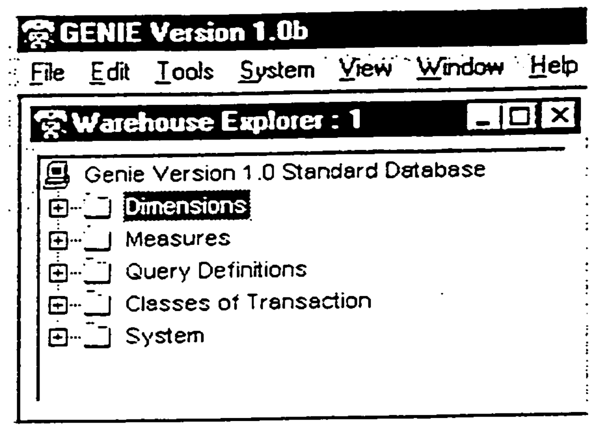


Fig. 17a

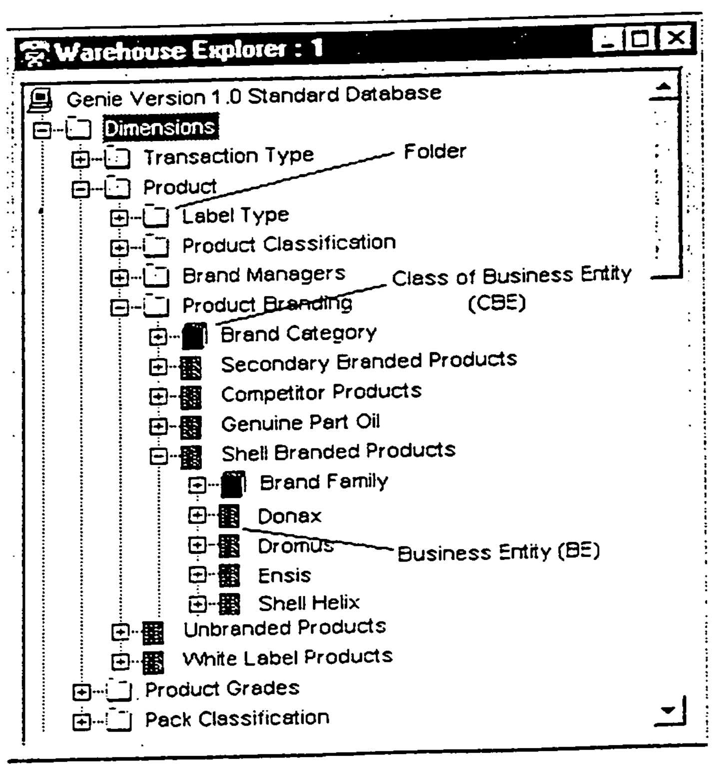


Fig. 17b

This Page Blank (usptc)

Fig. 18

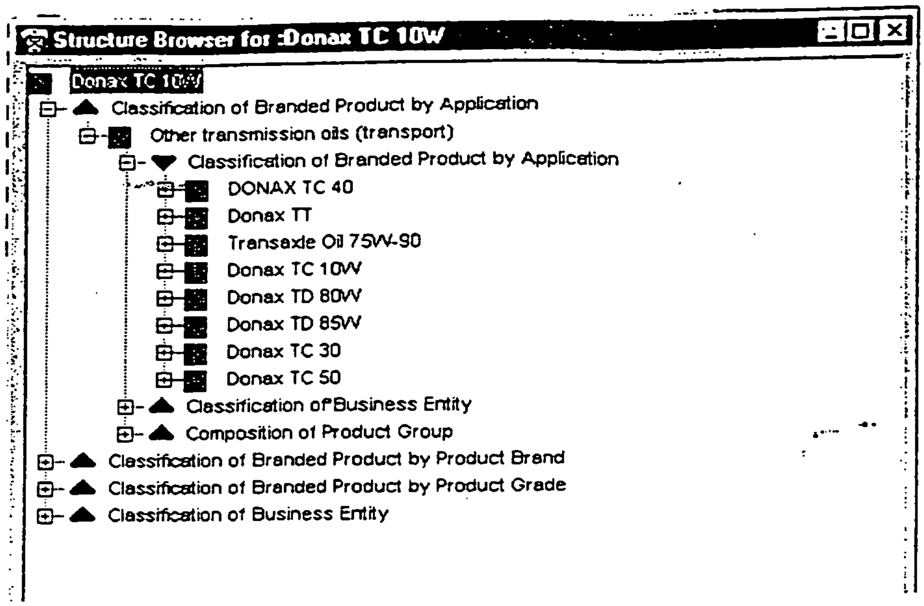
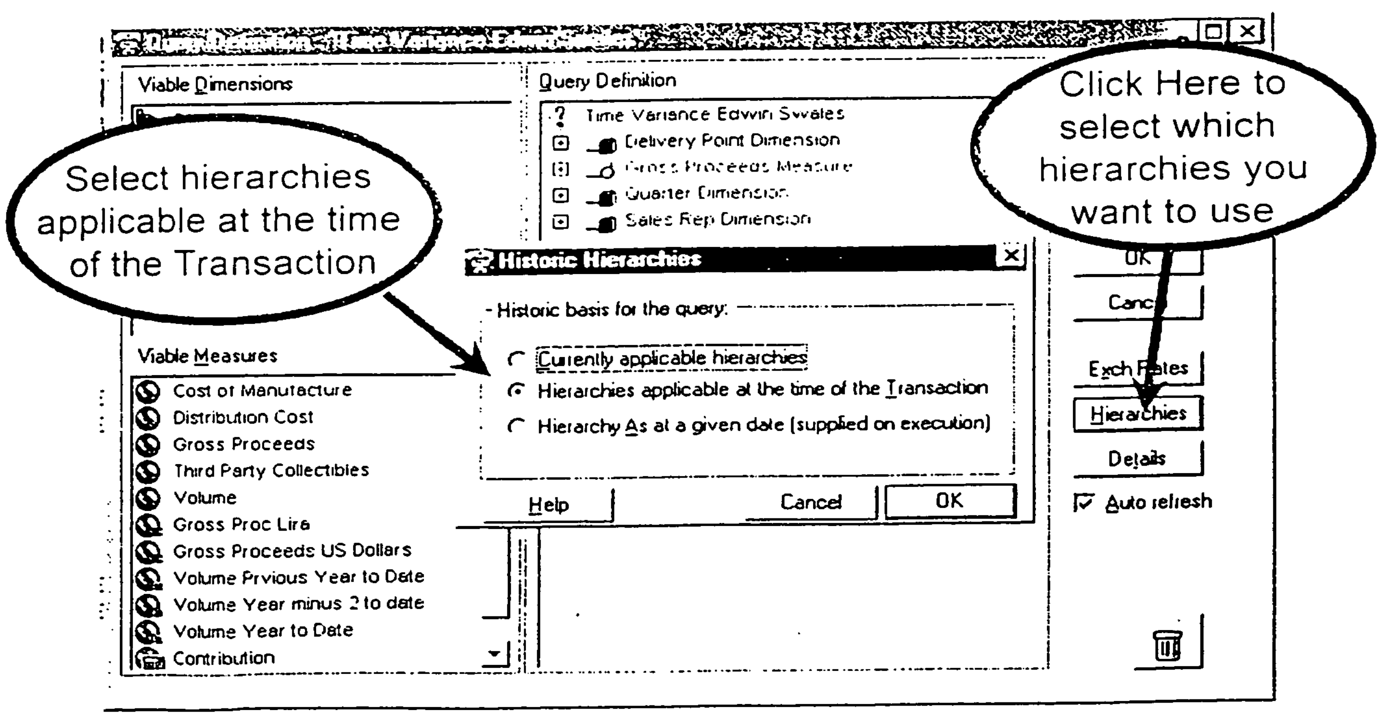

Structure Browser for :Donax TC 10W
Donax TC 10W
Classification of Branded Product by Application
Other transmission oils (transport)
Classification of Branded Product by Application
DONAX TC 40
Donax TT
Transaxle Oil 75W-90
Donax TC 10W
Donax TD 80W
Donax TD 85W
Donax TC 30
Donax TC 50
Classification of Business Entity
Composition of Product Group
Classification of Branded Product by Product Brand
Classification of Branded Product by Product Grade
Classification of Business Entity

Fig. 19


Viable Dimensions
Query Definition
Time Variance Edwin Swales
Delivery Point Dimension
Quarter Dimension
Sales Rep Dimension
Select hierarchies applicable at the time of the Transaction
Click Here to select which hierarchies you want to use
Historic Hierarchies
Historic basis for the query:
Currently applicable hierarchies
Hierarchies applicable at the time of the Transaction
Hierarchy As at a given date (supplied on execution)
Help
Cancel
OK
Exch Rates
Hierarchies
Details
Auto refresh
Viable Measures
Cost of Manufacture
Distribution Cost
Gross Proceeds
Third Party Collectibles
Volume
Gross Proc Lira
Gross Proceeds US Dollars
Volume Prvious Year to Date
Volume Year minus 2 to date
Volume Year to Date
Contribution

This Page Blank (uspto)

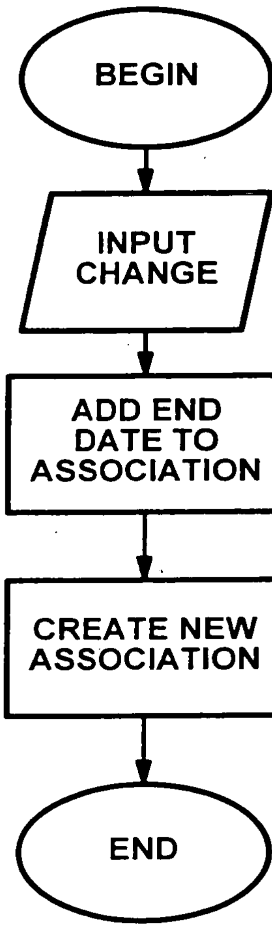

BEGIN
INPUT CHANGE
ADD END DATE TO ASSOCIATION
CREATE NEW ASSOCIATION
END

FIG. 21

This Page Blank (uspto)

This Page is Inserted by IFW Indexing and Scanning Operations and is not part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☐ FADED TEXT OR DRAWING
- ☐ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☐ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☒ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: ______________________

IMAGES ARE BEST AVAILABLE COPY.
As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.

This Page Blank (usp